

MICROHOBBY

AMSTRAD

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES AMSTRAD

Semanal

AÑO I N.º 18

160 Ptas.

Canarias 160 ptas.

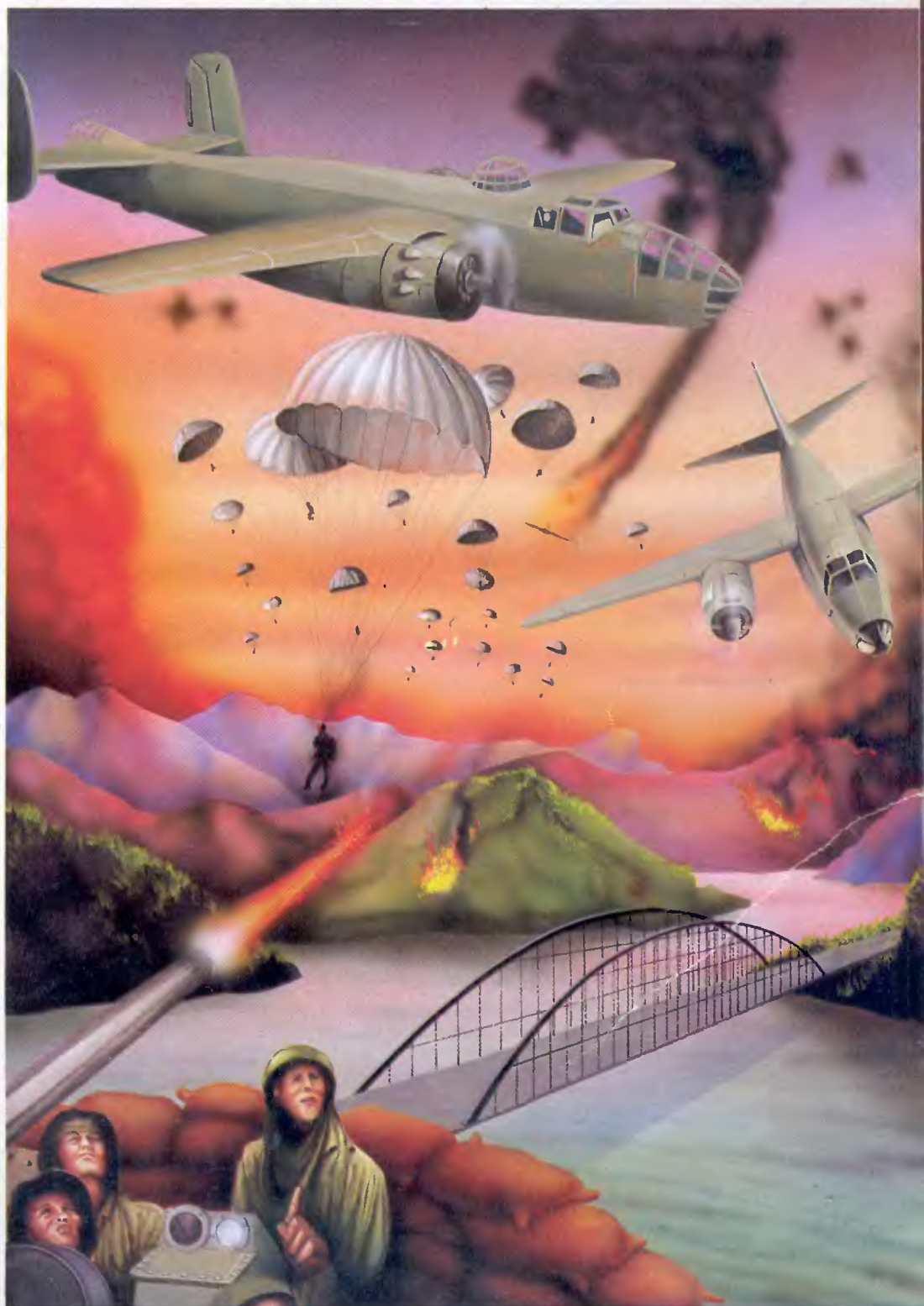
CALCULUS:
LA CALCULADORA
MÁS POTENTE
DEL MUNDO
ES UN AMSTRAD

LENGUAJES DE
INTELIGENCIA
ARTIFICIAL
Y QUINTA
GENERACION

CONTABILIDAD
GENERAL
CON PLACON
BAJO CP/M

SOFTWARE

La batalla
de Arnhem.
Las fuerzas
alemanas
defienden el
último puente



SACALE EL JUGO A TU ORDENADOR. DISEÑA TUS PROPIAS PANTALLAS Y DIVIERTETE JUGANDO CON...

Este mes:

YOUR COMPUTER

Te ofrece algo realmente sabroso:

MUSICA

Este magnífico programa escrito en código máquina te permitirá manejar el sonido y la música en tu Amstrad desde Basic, mediante un nuevo juego de comandos creados especialmente para ello.

CROSS

Tienes cuatro revólveres para destruir a tus enemigos en el mínimo tiempo posible. Necesitarás toda tu habilidad, rapidez de reflejos y suerte, mucha suerte.

JUMPER

Debes alcanzar la cima del Valle de las Cintas Deslizantes. Tienes que saltar por los huecos de las vallas, que se desplazarán a derecha e izquierda con una rapidez de vértigo.

MAGGOT

Te encuentras en la amable tierra de las setas gigantes. Tu misión es guardarla del ataque y la invasión de una peligrosísima serpiente polimórfica que las ataca sin piedad.

TIMEBOMB

Una organización terrorista de Oriente Medio ha colocado una bomba de tiempo en el laberinto de defensa del Laboratorio de investigación bacteriológica de Lexington.

RSX

Your Computer ha pensado en los usuarios del Amstrad CPC464 y ha creado un nuevo juego de comandos completo para tu ordenador, de forma que el Basic así ampliado no tenga nada que envidiar al de los otros modelos de la serie.

2

YOUR COMPUTER

EL CORAZON DE LA PRIMERA REVISTA EUROPEA DE ORDENADORES

AMSTRAD

464-664-6128

La mejor selección de programas de juegos y utilidades, publicados en la revista de mayor difusión de Europa en ordenadores. Ahora reproducidos en cassette, en auténtica exclusiva mundial.

695.- PTAS.

1 Cross
2 Jumper
3 Maggot
4 Timebomb
5 Música
6 RSX

Las cintas de Your Computer se editan mensualmente para Amstrad, Spectrum y Commodore.

¡ESTA VENTA EN TU KIOSCO!

¡GANA UN 128 K!

Total garantía de carga

SMITH & SONS

Si no lo encontrara en su kiosco, puede solicitarlo directamente a nuestra editorial:
Paseo de la Castellana, 268. Tel.: (91) 733 25 99. 28046 Madrid.

MICROHOBBY

AMSTRAD

sumario

Año 1 • Número 18 • 31 de Diciembre de 1985 al 6 de Enero de 1986
150 ptas. (sobretasa Canarias, 10 ptas.)

Director Editorial

José I. Gómez-Centurión

Director Ejecutivo

Víctor Prieto

Subdirector

José María Díaz

Redactora Jefe

Marta García

Diseño

José Flores

Colaboradores

Francisco Portalo, Pedro Sudón

Miguel Sepúlveda,

Francisco Martín,

Jesús Alonso, Pedro S. Pérez

Amalio Gómez

Juan J. Martínez,

David Sopuerta, Alberto Suárez,

Eduardo R. Velasco

Secretaría Redacción

Carmen Santamaría

Fotografía

Carlos Candel

Javier Martínez

Portada

M. Barco

Ilustradores

Javier Igual, J. Pons, F. L.

Frontón, J. Septien, Pejo, J. J.

Mora, Luigi Pérez

Edita

HOBBY PRESS S.A.

Presidente

Mario Andriño

Consejero Delegado

José I. Gómez-Centurión

Jefe de Publicidad

Concha Gutiérrez

Publicidad Barcelona

José Galán Cortes

Tel: (93) 303 10 22/313 71 62

Secretaría de Dirección

Marisa Cogorro

Suscripciones

M.ª Rosa González

M.ª del Mar Calzada

Redacción, Administración y Publicidad

La Gröña, 39

Polígono Industrial de Alcobendas

Tel.: 654 32 11

Telex: 49 480 HOPR

Dto. Circulación

Carlos Peropadre

Distribución

Coedis, S. A. Valencia, 245

Barcelona

Imprime

ROTEDEC, S. A. Ctra. de Irún.

Km. 12,450 (MADRID)

Fotocomposición

Novocomp, S.A.

Nicolás Morales, 38-40

Fotomecánica

GROF

Ezequiel Solano, 16

Depósito Legal:

M-28468-1985

Derechos exclusivos

de la revista

**COMPUTING with
the AMSTRAD**

Representante para Argentina, Chile,
Uruguay y Paraguay, Cia.
Americana de Ediciones, S.R.L. Sud
América 1.532. Tel.: 21 24 64. 1209
BUENOS AIRES (Argentina).

M. H. AMSTRAD no se hace
necesariamente solidaria de las
opiniones vertidas por sus
colaboradores en los artículos
firmados. Reservados todos los
derechos.

Se solicitará control OJD

5 Primera Plana

Semiconductores japoneses de 64 megabytes. Nueva interface RS232 para nuestros Amstrads.

6 Primeros pasos

Los ordenadores no sólo sirven para jugar, como muy bien saben todos nuestros lectores. También funcionan, y muy bien, a la hora de resolver aplicaciones que involucran una serie de cálculos complejos. El uso de las funciones trigonométricas servirá para complementar nuestro estudio de cómo el Amstrad trata los números, aparte de revelarse extremadamente útiles para los que vosotros mismos creéis.



12 Serie oro

El programa de Serie oro de esta semana presenta una aplicación de los ordenadores muy clásica y útil, pero que normalmente se olvida, porque se piensa que va implícitamente unida a una computadora: usar el Amstrad como una potentísima calculadora. Si alguien piensa que esto no es nada del otro mundo, que espere a ver la belleza con que el autor ha tratado la presentación de su obra y la sorprendente sencillez con que se maneja.

16 Análisis

Nunca está de más saber cómo el Amstrad organiza las coordenadas de la pantalla de texto. Por eso, Análisis ha preparado un programa que nos informa de ello sin ningún esfuerzo por nuestra parte. Sólo mueva el cursor.



Arnhem: 18

17 de septiembre de 1944. El primer war game en España para todos los modelos de Amstrad.

20 Lenguajes (y III)

En esta tercera y última parte de nuestra panorámica de los lenguajes de programación, tratamos de dar al lector una visión de conjunto, necesariamente superficial, de lo que son los lenguajes declarativos de Inteligencia Artificial, y de qué se cuece en la Quinta Generación de ordenadores.

23 Banco de pruebas

Placon es un programa de contabilidad muy potente, adaptado al Plan Contable Español, y su análisis aquí se justifica debido a la gran parte de usuarios a los que puede interesarles una aplicación tan especializada.

Código Máquina 28

Esta vez no hablamos de rutinas, sino de herramientas imprescindibles para crearlas. Al lector le sonará el concepto de programa ensamblador, pero cuando vea el ensamblador de Hisoft, que probamos esta semana, comprenderá por qué programar en máquina puede llegar a ser muy fácil.

COMPUTIQUE

64.900Ptas.

Amstrad 464 f.
verde

Te da más

GARANTIA **AMSTRAD**
ESPAÑA



Al comprar tu Amstrad te regalamos

• Estuche con ocho programas originales

- Fruit Machine
- Procesador texto
- Almirante Graf
- Oh Mummy
- Plaga Galáctica
- Amsdraw
- Laberinto Sultan
- Animal, Vegetal, Mineral

• Joystick Gunshot I

• Un estupendo libro de Basic

• Los cuatro mejores programas:

- Decathlon
- Jet Set Willy
- Sabrewulf
- Beach-Head

• Guía de referencia del programador

• y además te obsequiamos con un curso de introducción al Basic.

VENTA A PLAZOS HASTA 36 MESES



Nuevo Amstrad CPC6128: 109.500 ptas. (F. Verde)

COMPUTIQUE

Servimos a tiendas
Abrimos sábados por la tarde

Embajadores, 90 Tfno. 2270980
28012 Madrid

Semiconductores de 64 megas

En Japón, en esto de la microelectrónica, no se duermen en los laureles en absoluto. No les basta con ser el país puntero, sino que sus esfuerzos de investigación no cesan en ningún momento (probablemente por eso son los primeros).

Un minúsculo sincroton, de 5 metros de diámetro, permitirá fabricar semiconductores de 64 megaoctetos (64.000 Kbytes) a partir de 1988.

Por otra parte, conectando con otra noticia publicada no hace mucho en **AMSTRAD** Semanal, el profesor Isao Karube, de la Universidad de Tokio, acaba de reivindicar la paternidad de la idea de fabricar «**bio-semiconductores**», usando materiales biológicos más que inorgánicos, con lo que la capacidad de integración superaría en mil millones de veces a la conseguida actualmente.

ORDENADORES Y BIBLIOTECAS

Desde hace algún tiempo existen en Cataluña unas cuantas bibliotecas que utilizan ordenadores como herramienta cultural interactiva con sus consultantes.

Las llamadas «**microtecas**» cuentan con unos cuantos ordenadores integrados en una red local, esto es, existe una estación central dotada de disco duro a la que acceden el resto de las máquinas para encontrar la información solicitada.

Esto libera al consultante de tener que andar sacando y metiendo disquetes, optimiza los recursos totales de la biblioteca y hace la intervención del bibliotecario necesaria solamente en los casos verdaderamente importantes.

Felicitaciones por una iniciativa que, si se dirige racionalmente, facilitará el acceso personalizado a la información sin incómodos estados de espera y liberará a unas cuantas personas de trabajos mecánicos y, por tanto, inhumanos.



Hogar por ordenador

Los ordenadores amenazan con introducirse muy profundamente no sólo en nuestro trabajo, sino también en nuestros hogares, tomando control de muchas funciones hasta ahora reservadas a nosotros.

En Japón, cómo no, usted puede agenciarse un aspirador del tamaño de un paquete de Ducados para limpiar las migas de la mesa antes de tomar el postre. Sí, sí, como suena.

Esto en sí mismo no es muy preocupante, pero en Francia se ha ideado un tipo de habitación pensada inicialmente para ayuda a los parapléjicos, que permite controlar mediante la voz cosas tales como bajar y subir persianas, descolgar un teléfono o hacer que un robot le acerque a uno un vasito de agua.

En este sentido, tal vez lo más espectacular sea la comercialización de una columna compuesta por un televisor y un teléfono: por ejemplo, llega un visitante a casa y el ordenador pasa la imagen al televisor. Usted, mediante el teléfono, o habla con el visitante o le ordena a la computadora que abra la puerta. El programa incluido en la máquina es capaz de detectar fugas de gas, incendios o la presencia de intrusos, y actuar en consecuencia.

En Silicon Valley, una investigadora francesa ha creado KATALAVOX, un sistema que permite controlar cualquier proceso que un ordenador sea capaz de hacer mediante la voz humana.

Será posible ir dictándole al ordenador un escrito mientras uno pasea por su casa, y en mitad de la narración, ordenarle cualquier otra cosa que se nos ocurra.

Los ordenadores estarán muy pronto en nuestra casa, para ayudarnos como mayordomos según unos, y para deshumanizarnos e incordiar en general, según otros. Veremos...

Primera plana



NUEVA RS232 PARA AMSTRAD

M.H.T. Ingenieros ha lanzado una interface serie RS232 que permite a cualquier modelo de la serie CPC comunicarse con plotters, impresoras y otros ordenadores remotos o no a través de modems.

La interface se conecta en el bus de expansión del **Amstrad**, y nos facilita para conectar más periféricos a través suyo, como un controlador de disco.

Viene con su propio software en una EPROM; no ocupa memoria del usuario y crea unos nuevos comandos Basic a través de RSX para gestionar la interface sin problemas.

El manual señala que la RS232 funcionará también a bajo CP/M 2.2 y CP/M 3.0 normalmente, después de un proceso de instalación. Por último, como detalle interesante, el manual explica un poco por encima cómo conectar con un Sinclair QL y un Spectrum.

HA LLEGADO LA TRIGONOMETRIA

La Trigonometría ha sido y será uno de los «huesos» más grandes que nos ha tocado «roer» al estudiar Matemáticas. Nos ha llenado la cabeza de fórmulas, teoremas, hipótesis, tesis... En fin, una locura. Seguro que si Pitágoras y cía hubieran tenido un ordenador a mano, la cosa hubiera sido más divertida y menos árida.

Javier Igual

8

En más preámbulos vamos a entrar en materia hablando de los ángulos, como siempre se ha hecho. Original idea, ¿verdad?

Lo primero que nos encontramos al hablar de ángulos son sus unidades de medida: radianes y grados.

Radianes y grados

¿Sería conveniente recordar la definición de radián?

Seguramente que a todos nos vendría bien refrescar la memoria: Radián es el ángulo cuyo arco de circunferencia tiene la misma longitud que el radio de ésta.

Si haciendo un **esfuerzo** decimos que la longitud de la circunferencia es:

$$\text{long} = 2 * \pi * \text{radio}$$

dividiendo por el radio obtendremos el número de radianes que hay en una circunferencia ($2 * \pi * \text{radio}$). Como ésta tiene 360 grados, si queremos encontrar el valor exacto en grados de un radián bastaría que tecláramos:

$$\text{radián} = 360 / (2 * \pi)$$

y después

PRINT radián

No es nuestra intención montar aquí toda una teoría sobre el cálculo de radianes y otras zarandajas. Solamente intentamos refrescarle un poco la memoria.

Para trabajar con cualquiera de

las funciones trigonométricas que vamos a ver a continuación es necesario que le digamos al **Amstrad** qué unidades vamos a manejar. Como sólo hay dos tipos no nos resultará muy difícil.

Ordenes DEG y RAD

Mediante el comando DEG le decimos que las unidades de todas las funciones trigonométricas que van a continuación están expresadas en grados. Sin embargo, si utilizamos RAD le estamos indicando que las tome como radianes.

Por si no se cree lo que le decimos, vamos a proponerle un ejemplo práctico. Escriba:

RAD

seguido de

PRINT SIN (30)

En la pantalla aparecerá el valor -0.988031625 . A continuación ponga:

DEG

y como antes

PRINT SIN (30)

El resultado será ahora un valor diferente al de antes. El número de unidades es el mismo —treinta— pero, en un caso son **radianes** y en el otro **grados**. ¡Curioso!

Advertencia. Si no le indicamos nada, el ordenador toma implícitamente como unidades los radianes. No se olvide de ello.

Podemos intentar hacer un pequeño programa que nos convierta radianes en grados. Antes de continuar inténtelo usted mismo.

$\text{Sen}^2 \hat{A} +$



$$\cos^2 \hat{A} = 1$$

El método es muy sencillo. Primero calculamos los grados que corresponden a los radianes del dato (líneas 40 y 50). Utilizamos para ello la función INT. A continuación calculamos el resto de esta división. Multiplicamos este valor por 60 y calculamos los minutos (línea 70). De un modo análogo podemos calcular los segundos.

Pero entremos de lleno dentro del mundo de la Trigonometría.

Función SIN

La primera función que vamos a ver es el seno de un ángulo: SIN —ya la hemos utilizado para el primer ejemplo de este artículo.

Su misión es calcular el seno del número o de la expresión numérica que va entre paréntesis.

Pruebe obtener los valores de 45 grados y de $\pi/2$ radianes.

¿Ya lo ha intentado? Confiamos que lo haya hecho de la siguiente forma:

```
DEG
PRINT SIN (45)
```

para obtener el primer valor, y:

```
RAD
PRINT SIN (PI/2)
```

para obtener el segundo. Fácil, ¿no?

Problemos emplear esta función también en sentencias de asignación. Podíamos repetir el ejemplo anterior de la siguiente forma. Teclee:

```
DEG
seno = SIN (45)
PRINT seno
```

o

```
RAD
seno = SIN (PI/2)
PRINT seno
```

y en ambos casos obtendremos los mismos valores que ya teníamos.

Con el Programa II podemos ver los diferentes valores que nos devuelve esta función para distintos ángulos con un intervalo de 30 grados.

Vemos que en el intervalo de 0 a 180 grados, el valor del seno es positivo, mientras en el resto de los ángulos de una circunferencia es negativo.

Función COS

Otra de las funciones trigonométricas de las que está provisto nuestro **Amstrad** es COS. Con ella podemos calcular el valor del coseno de

Primeros pasos

la expresión numérica o del número que figura entre paréntesis.

Como ya se vio para la función SIN, también en este caso hay que decirle previamente en qué tipo de unidades vamos a darle el ángulo.

Pruebe a teclear:

```
DEG
```

seguido de

```
PRINT COS (45)
```

y obtendremos el mismo valor que haciendo:

```
RAD
PRINT COS (PI/4)
```

ya que los ángulos de 45 grados y $\pi/4$ radianes son iguales.

En el programa 3 podemos ver los diferentes valores que va tomando la función COS para ángulos comprendidos entre 0 y 360 grados tomados en intervalos de 30 en 30.

Observa que estos valores son positivos entre 0 y 90 grados, negativos entre 90 y 270, y nuevamente positivos en el intervalo de 270 a 360 grados.

A semejanza de SIN, podemos utilizar también la función COS en sentencias de asignación. Compruebe que es válido teclear:

```
RAD
coseno = COS (PI/2)
```

seguido de

```
PRINT coseno
```

La tangente de un ángulo es la razón existente entre el seno y el coseno del mismo. ¡Bonita y tradicional definición! Si los clásicos hubiesen tenido un ordenador hubieran realizado el cálculo de la tangente de, por ejemplo 60 grados, del modo siguiente:

```
DEG
seno = SIN (60)
coseno = COS (60)
tangente = seno/coseno
```

y de haber querido conocer su valor harían:

```
PRINT tangente
```


Función TAN

Con el **Amstrad** y nuestra mentalidad informática, la cosa es más sencilla. Esta maravillosa máquina está dotada de una función que nos ahorra hacer las operaciones matemáticas. La forma de calcular la tangente sería:

```
DEG
PRINT TAN (60)
```

y asunto terminado. En la pantalla tendremos el **flamante** valor de la tangente de 60 grados.

Podemos preparar un programita para conocer los valores por los que va pasando la tangente al recorrer el ángulo una circunferencia completa (360 grados o 2π radianes).

Le invitamos a que se atreva a fabricarse este programita antes de ver la solución que nosotros le proponemos. También nos atrevemos a sugerirle que estudie ciertos casos concretos que se le van a dar con alguno de los valores obtenidos. **¿Cómo lo ve?**

Si después de intentarlo quiere conocer nuestra solución, basta con que eche una ojeada al programa 4. No le será difícil entenderlo.

La única novedad que tiene es el tratamiento que hacemos a los casos de error en las líneas 120 y 130. En los puntos en los que hemos detectado error éste se ha producido debido a que el valor de la tangente es o muy grande o muy pequeño. Nosotros hemos generalizado y determinado, mediante el tratamiento de error, que este valor sea INFINITO.

Observe que estos casos se dan en ángulos muy próximos a $\pi/2$ ó $3\pi/2$ radianes.

Solamente nos queda por ver la función encargada de calcular el arco cuya tangente es la expresión numérica que está encerrada entre paréntesis. Su forma general es:

$\arctg = \text{ATN} (\text{expresión})$

En este caso, ¡cómo no!, podemos hacer también que la función nos devuelva el arco en grados o en radianes.



Compruébelo con:

```
DEG
PRINT ATN (1)
```

seguido de

```
RAD
PRINT ATN (1)
```

¿Verdad que comprende el porqué de la diferencia de los dos valores obtenidos?

Intente encontrar los valores del arco para tangentes de muy distintos rangos. Generalmente los casos límites son los que no hemos previsto y los que nos dan muchos problemas inesperados.

Nos va a volver a llamar pesados: Analice si el valor numérico devuelto por todas estas funciones está dentro de los límites correctos. Si no es así, trataremos el error y daremos el mensaje correspondiente.

Bueno, ya hemos visto todas las funciones trigonométricas que nos ofrece nuestro ordenador. Conociendo el manejo de las mismas y las fórmulas y teoremas de los antiguos podemos ser capaces de resolver cualquier triángulo que nos pongan delante.

Más interesante que el manejo teórico de estas funciones es su aplicación concreta a problemas prácticos que encontramos en la vida corriente. Por ejemplo, un topógrafo sería feliz teniendo un programa que usara estas funciones y le resolviera el problema de calcular alturas de edificios o elevaciones de terreno.

A nuestro topógrafo le bastaría con medir la distancia que hay desde el punto donde se encuentra hasta el edificio y el ángulo comprendido entre la horizontal y la línea que une dicho punto y la azotea (*parte más alta de la casa*). Si sigue el programa V verá por qué.

Se trata de un algoritmo de **andar por casa** que nos realiza este cálculo.

Con las líneas 40 a 100 tomamos los datos que nos proporciona el topógrafo y los colocamos en sus variables correspondientes.

En la línea 110 realizamos el cálculo de la altura. Si no hay cosas extrañas en los datos, imprimimos el resultado (línea 120).

Si no va bien la cosa, sacamos el mensaje de error (*haga caso de la sugerencia*) y le pedimos nuevos datos para calcular la altura.

De momento vale por hoy. La próxima semana seguiremos nuestro paseo por el mundo de los números. ¡Ah!, prepárese para ser MILLONARIO.

PROGRAMAS

```
10 REM PROGRAMA I
20 CLS
30 INPUT "Radianes: ",radianes
40 total=radianes*360/(2*PI)
50 grados=INT(total)
60 resto1=(total-grados)*60
70 minutos=INT(resto1)
80 resto2=(resto1-minutos)*60
90 segundos=INT(resto2)
100 PRINT
110 PRINT "GRADOS: "grados, "MINUTOS: "minutos, "SEGUNDOS: "segundos
```

```
10 REM PROGRAMA II
20 CLS
30 DEG
40 PRINT "ANGULO: ", "SENO"
50 PRINT "-----", "-----"
60 FOR angulo=0 TO 360 STEP 30
70 seno=SIN(angulo)
80 PRINT angulo, seno
90 NEXT angulo
```

```
10 REM PROGRAMA III
20 CLS
30 DEG
40 PRINT "ANGULO: ", "COSENO"
50 PRINT "-----", "-----"
60 FOR angulo=0 TO 360 STEP 30
70 coseno=COS(angulo)
80 PRINT angulo, coseno
90 NEXT angulo
```

```
10 REM PROGRAMA IV
20 ON ERROR GOTO 120
30 CLS
40 DEG
50 PRINT "ANGULO: ", "TANGENTE"
60 PRINT "-----", "-----"
70 FOR angulo=0 TO 360 STEP 30
80 tangente=TAN(angulo)
90 PRINT angulo, tangente
100 NEXT angulo
110 END
120 PRINT angulo, " INFINITO"
130 RESUME 100
```

```
10 REM PROGRAMA V
20 ON ERROR GOTO 140
30 CLS
40 INPUT "DISTANCIA HORIZONTAL: ",distancia
50 IF distancia=0 THEN ERR=11
60 PRINT
70 INPUT "GRADOS O Radianes (g/r).",unidad$
80 IF unidad$="g" THEN DEG ELSE IF unidad$="r" THEN RAD ELSE GOTO 70
90 PRINT
100 INPUT "ANGULO: ",angulo
110 altura=distancia*TAN(angulo)
120 PRINT altura
130 END
140 REM TRATAMIENTO DE ERROR
150 PRINT
160 PRINT "ESTAS DEBAJO DEL EDIFICIO . SAL CORRIENDO POR SI ACASO SE DERRUMBA."
170 FOR i=0 TO 3000:NEXT i
180 RESUME 30
```


MICRO-1

Duque de Sexto, 50. 28012 Madrid
Tels. (91) 274 53 80-276 96 16



SOFTWARE: ¡¡GRATIS 1 BOLIGRAFO DE ACERO CON RELOJ INCORPORADO!!

	Ptas.		Ptas.
Bounty Bob	2.300	Bruce Lee	2.300
Fighting Warrior	2.100	Yier Kung Fu	2.300
Foupack	3.890	Exploding Fist	2.300
Combat Lynx	2.100	Southern Belle	2.300
Dummy Run	2.100	Dragontorc	2.300
Raid	2.300	Rocky	2.100
Match Day	2.300	World Series Baseball	2.100
Map Game	2.750	Dambusters	2.300
Hypersports	2.300	Ajedrez Tridimensional	1.975

IMPRESORAS: ¡¡20% DE DESCUENTO SOBRE P.V.P.!!

Lápiz óptico DK'Tronics	4.850	Diskette 3"	1.050
Tapa metacrilato AMSTRAD	1.975	Cinta C-15 (especial)	85
Toshiba MSX 64 K	39.900	Cassette Especial	5.295

Joystick Quick Shot II
2.495 ptas.

Joystick Quick Shot I
1.995 ptas.

Joystick Quick Shot V
2.995 ptas.

Increíbles precios para tu AMSTRAD
464 y 6128 (verde y color).
(Llámanos y te asombrarás)

PC Compatible IBM 256 K
Monitor Fósforo Verde
2 Bocas Diskette 360 K
279.000 ptas.

Sabrewulf + Decathlon + Beach
Head + Jet Set Willy
2.500 ptas.

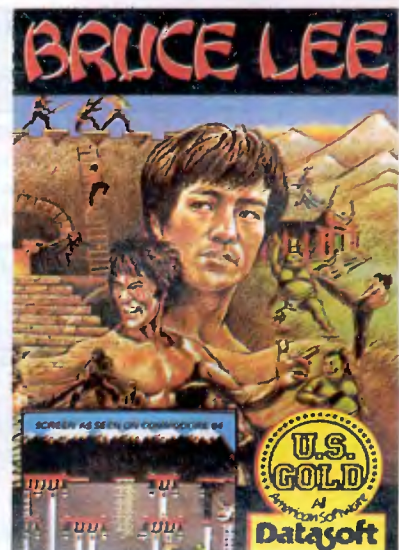
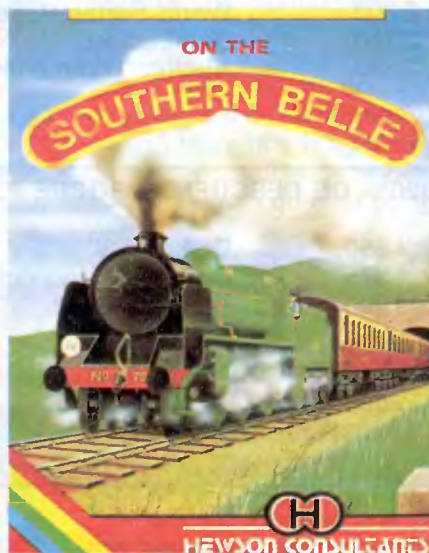
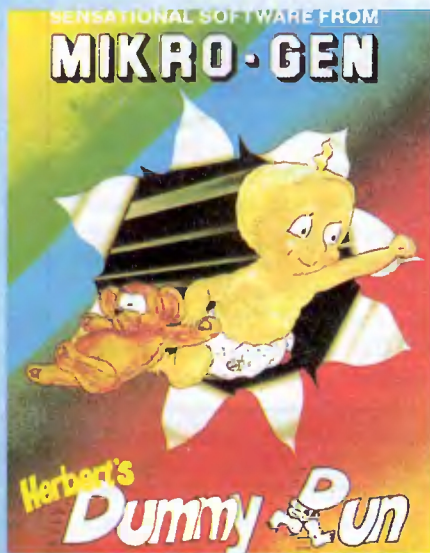
Commodore 64: 42.900 ptas.
Commodore 128: 74.900 ptas.

El pedido te lo enviamos URGENTEMENTE contra-reembolso SIN NINGUN GASTO DE ENVIO, LLA-MANDO a los teléfonos: (91) 276 96 16-274 53 80 o escribiendo a MICRO-1. Duque de Sexto, 50. 28012 Madrid.



PREPA

Dis



SI BUSCAS LO MEJOR **ERBE** Software LO TIENE

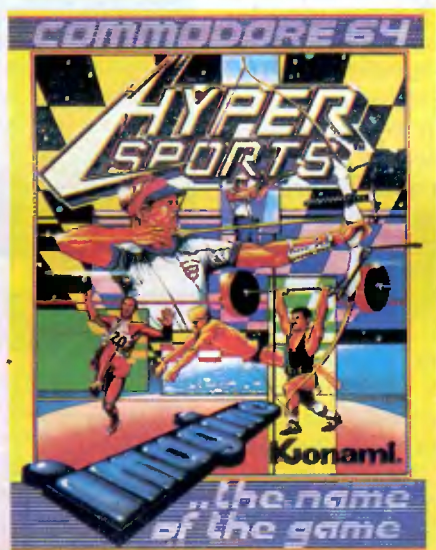
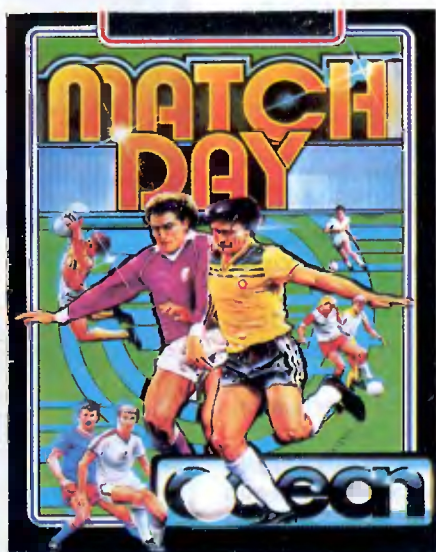
PARA Las MALETAS A

Disney World!



Software

TE INVITA



COMO LO OYES. EN CADA UNO DE NUESTROS JUEGOS ENCONTRARAS UN CUPON. MANDANOSLO DEBIDAMENTE RELLENADO Y PARTICIPARAS EN EL SORTEO QUE ENTRE TODOS LOS CUPONES RECIBIDOS HAREMOS EL 15 DE ENERO. ¡IMAGINATE! NADA MENOS QUE VIAJE Y ESTANCIA PARA 2 PERSONAS DURANTE 1 SEMANA EN MIAMI, EN DISNEY WORLD.

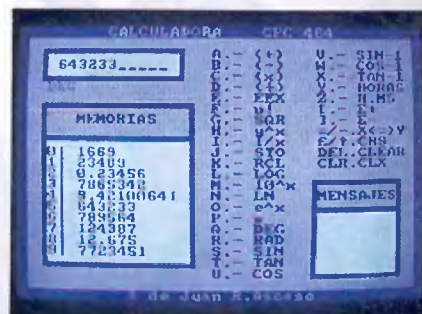
STA. ENGRACIA, 17 - 28010 MADRID. TEL. 447 34 10

CALCULADORA CPC 464

Cabe preguntarse, ¿por qué limitar un ordenador a operar como una calculadora, cuando podemos realizar operaciones de forma directa con un resultado inmediato? (ejemplo: PRINT 3-5). La respuesta es muy sencilla, siempre que nos limitemos a operar con ciertos valores y olvidarlos, no hay problema. Pero cuando deseamos guardar esos valores y operar con ellos de forma estadística necesitamos ya de un pequeño programa para llevarlo a cabo. Con esta calculadora podemos realizar todo esto y más.

Programa realizado por el lector:

JUAN RAMON ASCASO AGUADO (Zaragoza)



E

l funcionamiento está basado en las calculadoras HEWLETT-PACKARD que utilizan para operar dos registros 'x' e 'y', en los cuales se introducen los valores y a continuación se opera con ellos. Se entenderá mejor con un ejemplo.

En una calculadora convencional sería:

5 + 3 = pantalla 8

en la calculadora HP:

5 ENTER 3 + pantalla 8

y en la calculadora CPC 464:

5 ENTER 3 A pantalla 8

La tecla A sería la que realizaría la suma de los registros, véase el menú de opciones.

En los dos últimos casos lo que se ejecuta es lo siguiente:

— introducimos el número 5 en el registro 'x'.

— pulsamos ENTER, hace una copia del registro 'x' en el registro 'y'.

— introducimos el número 3 en el registro 'x' borrando el número anterior (5).

— opera los registros con la operación indicada y deja el resultado en el registro 'x'.

MANEJO

Comprendido cómo funciona el ejemplo anterior el resto de operaciones no resulta nada complicado, al contrario es de muy fácil manejo.

Al 'encender' la calculadora (RUN), aparecen en la pantalla tres ventanas informativas, un menú de opciones y un indicador intermitente del sistema de grados.

La ventana superior izquierda, a la que llamaremos pantalla 1, es la que nos muestra el número que estamos entrando en el registro 'x' o el resultado de las operaciones.

La ventana más grande, que llamaremos pantalla 2, nos muestra las memorias almacenadas o la posibilidad de visualizar los datos estadísticos. Y la ventana derecha, que será la pantalla 3, nos aparecen los mensajes informativos o de error.

Tenga en cuenta que siempre que opere con dos números, debe de separarlos con la tecla ENTER y para finalizar la operación a realizar, que puede ser una suma, un producto, la potenciación, etc.

Y para operaciones que sólo necesitan un operando, se introduce el número y seguidamente la tecla de la función, como son la raíz, la inversa, factorial, seno, etc.

Siempre que deseemos operar con una constante deberá encontrarse en el registro 'y', ya que ésta no se borra hasta que volvamos a pulsar ENTER, que copiará en el registro 'y' lo que contenga el registro 'x'.

ESTRUCTURA

10-	50	Presentación autor
	60	Detección de errores
70-	80	Define colores
	90	Define caracter
100-	130	Programa teclas
	140	Dimensionar e inicializar variables
150-	360	Presentación de pantalla
	370	Define ventana mensajes
380-	400	Desvío a subrutinas
410-	470	Bucle principal de entrada de datos
480-	680	Subrutinas varias (1)
690-1.690		Subrutinas de operaciones (2)

SUBROUTINAS VARIAS

510-	560	Compone número
570-	580	Avisa error
590-	600	Borra registro 'x'
610-	630	Borra todas las variables
640-	660	Saludo
670-	680	Imprime resultado en registro 'x'

Por ejemplo, si deseamos sumar la misma cantidad (25) a una serie de números lo haremos así:

25 **ENTER** 7 **A** pantalla 132
 25 **ENTER** 9 **A** pantalla 134
 25 **A** pantalla 150

MEMORIAS

En la memoria sólo podemos almacenar el número que contenga el registro 'x'. Si queremos almacenar el número del registro 'y' deberemos intercambiar el contenido de los registros y a continuación almacenar el dato.

Si deseamos almacenar el 32.570 en la memoria 6 lo haremos así:

32.570 **J** pantalla
 pantalla 3
6 pantalla 2
 32.570

Guardar en memoria n.º ?
 lugar correspondiente a la memoria 6 tendremos el 32.570.

Para almacenar un dato del registro 'y', el 95.237 en la memoria 2 se realizará:

95.237 **ENTER** **CLR** pantalla 1
= pantalla 1
J pantalla 3
2 pantalla 2
 95.237

Guardar en memoria n.º ?
 Lugar correspondiente a la memoria 2 tendremos el 95.237.

Y para obtener un dato almacenado de la memoria, pulsaremos la opción **K** y el número de memoria en que se encuentre el dato que deseamos, el cual será copiado en el registro 'x'.

ESTADISTICA

La calculadora CPC 464 nos permite también introducir una serie de datos para obtener la media aritmética o la desviación estándar.

Supongamos que hemos obtenido en un problema unos valores correspondientes a los meses del año 1984 y otros del año 1985. Deseamos comparar la media de cada año y la desviación estándar.

Valores de 1984: (80, 21, 75, 42, 65, 73, 92, 32, 52, 54, 77, 30)
 valores de 1985: (73, 32, 51, 37, 66, 84, 95, 50, 89, 71, 49, 57)

Ahora debemos decidir en qué registro va a ir cada año. Supongamos que el año 1984 lo introducimos en el registro 'y' y el año 1985 en el registro 'x'. Se introduciría así:

80 **ENTER** 73 **I** pantalla 1
 pantalla 2
 21 **ENTER** 32 **I** pantalla 1
 75 **ENTER** 51 **I** pantalla 1
 42 **ENTER** 37 **I** pantalla 1
 65 **ENTER** 66 **I** pantalla 1
 73 **ENTER** 84 **I** pantalla 1
 92 **ENTER** 95 **I** pantalla 1
 32 **ENTER** 50 **I** pantalla 1
 52 **ENTER** 89 **I** pantalla 1
 54 **ENTER** 71 **I** pantalla 1
 77 **ENTER** 49 **I** pantalla 1
 30 **ENTER** 57 **I** pantalla 1
 pantalla 2

0 12
 1 754
 2 51872
 3 62.8333333
 4 20.2162551
 5 693
 6 45721
 7 57.75
 8 22.7641065
 9 46598

NOTA: Cuando se emplea la memoria para datos estadísticos no se pueden sacar los resultados de la memoria, si no que deberá copiarlos por teclado para después almacenarlos si lo desea.

Habrás observado que en la pantalla 2 se nos muestran unos datos correspondientes a los valores que hemos introducido. La interpretación es la siguiente:

MEMORIAS	RESULTADO
0	números de datos introducidos en cada registro. (re)
1	suma los registros 'x'. (Σx)
2	suma los cuadrados de los registros 'x' (Σx^2)
3	media aritmética de los registros 'x' ($\Sigma x/re$)
4	desviación estándar de los registros 'x' ($S_x = \frac{re \cdot xe^2 - Xe^2}{re \cdot (re-1)}$)
5	suma los registros 'y'. (Σy)
6	suma de los cuadrados de los registros 'y' (Σy^2)
7	media aritmética de los registros 'y' ($\Sigma y/re$)
8	desviación estándar de los registros 'y' ($S_y = \frac{re \cdot ye^2 - ye^2}{re \cdot (re-1)}$)
9	suma de los productos del registro 'x' por el registro 'y'. (Σxy)

Serie Oro

Si cualquiera de los valores introducidos los deseamos modificar sin tener que meter todos de nuevo, lo haríamos como sigue:

Supongamos que desamos modificar el penúltimo dato de entrada que eran 77 y 49 por los nuevos 55 y 72.

77 **ENTER** 49 **I** pantalla 1 0
 55 **ENTER** 72 **I** pantalla 1 0

Si ha terminado de trabajar con datos estadísticos y desea que aparezcan en la pantalla 2 los datos que había almacenado en la memoria, basta con pulsar **K** o **J**, tenga cuidado con esta última ya que le pedirá el número de memoria a guardar, imprimiendo lo que contenga en este momento el registro 'y' y le borrará el dato que contuviera anteriormente.

Por el contrario si lo que desea es poder ver en cualquier momento el contenido de la memoria estadística, asegúrese de que los dos registros 'x' e 'y' estén a cero, para ello pulse:

CLR **ENTER** **I**

Y POR ULTIMO...

También permite la calculadora pasar horas, minutos, y segundos a horas. Para ello debemos de separar las horas de los minutos y segundos con el punto decimal. Por ejemplo si deseamos pasar 20 h., 35 m., 9 seg. a horas introducimos:

20.3509 **Y** pantalla 1 20.5848333

Las últimas líneas de programa nos permiten elegir el color de nuestra calculadora tan sólo con pulsar la BARRA DE ESPACIO con unos colores ya fijados, que podrá modificar a su gusto.


```

E 30,12:PRINT"DEL.CLEAR":LOCATE 30,
13:PRINT"CLR.CLY"
350 LOCATE 30,15:PRINT STRING$(8,13
1):LOCATE 38,15:FOR a=1 TO 9:PRINT
CHR$(133);CHR$(8);CHR$(10);:NEXT a:
PRINT CHR$(11);CHR$(8);:PAPER 2:FOR
a=1 TO 8:PRINT CHR$(140);STRING$(2
,8);:NEXT a:PAPER 0:FOR a=1 TO 9:PR
INT CHR$(138);CHR$(8);CHR$(11);:NEX
T a
360 LOCATE 30,17:PRINT STRING$(8,14
0):LOCATE 30,16:PRINT"MENSAJES"
370 WINDOW 30,37,18,22:PAPER 2:PEN
3:CLS:REM ** ventana de mensajes **
380 GOSUB 650:REM ** saludo **
390 GOSUB 600:REM ** situar cifras
**
400 GOSUB 1110:REM ** grados **
410 *****
**
420 ***** ENTRADA DE TECLADO ****
**
430 *****
**
440 tecla$=UPPER$(INKEY$):IF tecla$
="" GOTO 440 ELSE IF NOT INKEY(18)
OR NOT INKEY(6) THEN y=x:GOSUB 680
450 ON INSTR("ABCDEFGHIJKLMNOPQRSTU
VWXYZ[]=-#~@! ",tecla$) GOSUB 730,7
50,770,790,810,820,850,870,890,910,
960,1010,1030,1050,1070,1090,1110,1
130,1150,1170,1190,1210,1230,1250,1
270,1290,1300,1350,1400,1400,1420,1
420,600,620,1440
460 IF (ASC(tecla$)>47 AND ASC(tecl
a$)<58) OR ASC(tecla$)=46 THEN IF w
=0 THEN GOSUB 600:w=1:GOSUB 520 ELSE
GOSUB 520
470 GOTO 440

```

A.— (—)	Suma
B.— (—)	Diferencia
C.— (x)	Producto
D.— (—)	División
E.— EEX	Exponente
F.— n!	Factorial
G.— SQR	Raíz
H.— y.X	Potenciación
I.— 1/x	Inversa
J.— STO	Almacena en memoria
K.— RCL	Recupera de memoria
L.— LOG	Logaritmo decimal
M.— 10.x	Antilogaritmo
N.— LN	Logaritmo natural
O.— eX	Antilogaritmo natural
P.— π	Valor de Pi
< espacio > .—	
Q.— DEG	Grados
R.— RAD	Radianes
S.— SIN	Seno
T.— TAN	Tangente
U.— COS	Coseno
V.— SIN-1	Inversa seno
W.— COS-1	Inversa coseno
X.— TAN-1	Inversa tangente
Y.— HORAS	Convierte a horas
Z.— H. M S	Convierte a h, m y s.
[.— Σ +	Acumula datos estadísticos
].— Σ —	Anula datos estadísticos
=/ —. — x \Leftrightarrow	
y	Intercambia registros
/ .— CHS	Cambio de signo
DEL.— CLEAR	Restaura calculadora
CLR.— CLX	Borra registro 'x'
Cambia color	

14 MICROHOBBY AMSTRAD

SUBROUTINAS DE OPERACIONES

720-730	Suma
740-750	Diferencia
760-770	Producto
780-790	División
800-810	Exponente
820-830	Factorial
840-850	Raíz cuadrada
860-870	Potenciación
880-890	Inverso
900-940	Almacena en memoria
950-990	Recupero de memoria
1.000-1.010	Logaritmo decimal
1.020-1.030	Antilogaritmo
1.040-1.050	Logaritmo natural
1.060-1.070	Antilogaritmo natural
1.080-1.090	Valor de PI
1.100-1.110	Grados (DEG)
1.120-1.130	Radianes (RAD)
1.140-1.150	Seno
1.160-1.170	Tangente
1.180-1.190	Coseno
1.200-1.210	Inversa seno
1.220-1.230	Inversa coseno
1.240-1.250	Inversa tangente
1.260-1.270	Convierte a horas
1.280-1.290	Convierte a horas, minutos y segundos
1.300-1.340	Acumula datos estadísticos
1.350-1.380	Anula datos estadísticos
1.390-1.400	Intercambio registros
1.410-1.430	Cambio de signo
1.440-1.690	Cambio de color

```

TO 9:LOCATE #2,1,1+a:PRINT#2,m$(a)
: NEXT a
920 tecla$=INKEY$:IF tecla$="" GOTO 920
930 IF ASC(tecla$)<48 OR ASC(tecla$)>57 GOTO 920
940 GOSUB 680:j=VAL(tecla$):m$(j)=x
x$:LOCATE #2,1,1+j:PRINT#2,"
":LOCATE #2,1,1+j:PRINT#2,m$(j):CLS:tecla$="" :RETURN
950 '===== K.- RCL =
=====
960 CLS:CLS#2:LOCATE 1,2:PRINT"DATO
DE MEMORIA No.?" :FOR a=0 TO 9:LO
CATE #2,1,1+a:PRINT#2,m$(a):NEXT a
970 tecla$=INKEY$:IF tecla$="" GOTO 970
980 IF ASC(tecla$)<48 OR ASC(tecla$)>57 GOTO 970
990 j=VAL(tecla$):x=VAL(m$(j)):GOSUB
680:CLS:tecla$="" :RETURN
1000 '===== L.- LOG10
=====
1010 x=LOG10(x):GOSUB 680:RETURN

```

```

1020 '===== M.- 10^X
=====
1030 x=10^x:GOSUB 680:RETURN
1040 '===== N.- LN
=====
1050 x=LOG(x):GOSUB 680:RETURN
1060 '===== O.- E^X
=====
1070 x=EXP(x):GOSUB 680:RETURN
1080 '===== P.- PI
=====
1090 x=PI:GOSUB 680:RETURN
1100 '===== Q.- DEG
=====
1110 DEG:LOCATE#3,4,6:PEN#3,3:PRINT
#3,"DEG":RETURN

```



P ara que los dedos
no realicen el trabajo duro, M.H. AMS
TRAD. In. ha. por h. Todas las listadas que incluyen
este logotipo se encuentran a tu disposición en un cas-
sette mensual, solicítaselo.

```

1120 '===== R.- RAD
=====
1130 RAD:LOCATE#3,4,6:PEN#3,3:PRINT
#3,"RAD":RETURN
1140 '===== S.- SIN
=====
1150 x=SIN(x):GOSUB 680:RETURN
1160 '===== T.- TAN
=====
1170 x=TAN(x):GOSUB 680:RETURN
1180 '===== U.- COS
=====
1190 x=COS(x):GOSUB 680:RETURN
1200 '===== V.- SIN-1
=====
1210 x=ATN(x/SQR(1-x^2)):GOSUB 680:
RETURN
1220 '===== W.- COS-1
=====
1230 x=ATN(SQR(1-x^2)/x):GOSUB 680:
RETURN
1240 '===== X.- TAN-1
=====
1250 x=ATN(x):GOSUB 680:RETURN
1260 '===== Y.- HORAS
=====
1270 x1=x-INT(x):x1=x1/60*100:x=INT
(x)+x1:GOSUB 680:RETURN
1280 '===== Z.- H.MS
=====
1290 x1=x-INT(x):x1=x1*60:x2=x1-INT
(x1):x2=x2*60:x=INT(x)+INT(x1)/100+
INT(x2)/10000:GOSUB 680:RETURN
1300 '===== ESTADISTICA +
=====
1310 IF x=0 AND y=0 THEN 1320 ELSE
re=re+1
1320 xe=xe+x:xe2=xe2+x^2:xmedia=xe/
re:ye=ye+y:ye2=ye2+y^2:xy=xy+x*y:y
media=ye/re:IF re>1 THEN Sx=SQR((re*
xe2-xe^2)/(re*(re-1))):Sy=SQR((re*y
e2-ye^2)/(re*(re-1)))
1330 CLS#2:LOCATE #2,1,1:PRINT#2,ST
R$(re):LOCATE #2,1,2:PRINT#2,STR$(
xe):LOCATE #2,1,3:PRINT#2,STR$(xe2
):LOCATE #2,1,4:PRINT#2,STR$(xmedi
a):LOCATE #2,1,5:PRINT#2,STR$(Sx):
1340 LOCATE #2,1,6:PRINT#2,STR$(ye)
:LOCATE #2,1,7:PRINT#2,STR$(ye2):
LOCATE #2,1,8:PRINT#2,STR$(ymedia):
LOCATE #2,1,9:PRINT#2,STR$(Sy):LO
CATE #2,1,10:PRINT#2,STR$(xy):x=0:
y=0:GOSUB 680:RETURN
1350 '===== ESTADISTICA -
=====
1360 IF (x=0 AND y=0) OR re<1 THEN
PRINT CHR$(7):RETURN
1370 re=re-1:xe=xe-x:xe2=xe2-(x^2):
xmedia=xe/re:ye=ye-y:ye2=ye2-(y^2):
xy=xy-x*y:ymedia=ye/re:IF re>1 THEN

```

Serie Oro

```

Sx=SQR((re*xe2-xe^2)/(re*(re-1))):
Sy=SQR((re*ye2-ye^2)/(re*(re-1))) E
LSE Sx=0:Sy=0
1380 GOSUB 1330:RETURN
1390 '===== X <=> Y
=====
1400 y1=y:x1=x:y=x1:x=y1:GOSUB 680:
RETURN
1410 '===== CHS
=====
1420 IF e=0 THEN x=-x:GOSUB 680:w=1
:RETURN
1430 IF x$(1)=" " AND chs=0 THEN te
cla$="-":GOSUB 520:chs=1:RETURN ELS
E PRINT CHR$(7):RETURN
1440 '===== COLOR
=====
1450 INK 0,26:INK 1,0:INK 2,13:INK
3,0,26
1460 A$=INKEY$:IF A$="" GOTO 1460
1470 IF A$<>" " THEN RETURN
1480 INK 0,25:INK 1,0:INK 2,26:INK
3,0,25
1490 A$=INKEY$:IF A$="" GOTO 1490
1500 IF A$<>" " THEN RETURN
1510 INK 0,20:INK 1,2:INK 2,26:INK
3,20,2
1520 A$=INKEY$:IF A$="" GOTO 1520
1530 IF A$<>" " THEN RETURN
1540 INK 0,11:INK 1,1:INK 2,26:INK
3,11,1
1550 A$=INKEY$:IF A$="" GOTO 1550
1560 IF A$<>" " THEN RETURN
1570 INK 0,19:INK 1,9:INK 2,26:INK
3,19,9
1580 A$=INKEY$:IF A$="" GOTO 1580
1590 IF A$<>" " THEN RETURN
1600 INK 0,16:INK 1,3:INK 2,26:INK
3,16,3
1610 A$=INKEY$:IF A$="" GOTO 1610
1620 IF A$<>" " THEN RETURN
1630 INK 0,0:INK 1,26:INK 2,13:INK
3,0,26
1640 A$=INKEY$:IF A$="" GOTO 1640
1650 IF A$<>" " THEN RETURN
1660 INK 0,17:INK 1,4:INK 2,26:INK
3,4,17
1670 A$=INKEY$:IF A$="" GOTO 1670
1680 IF A$<>" " THEN RETURN
1690 RETURN

```

VARIABLES

x\$(i)	Almacena n.º de entrada
tecla \$	Guarda tecla pulsada
i	Cuenta los números de entrada
xx\$	Almacena el número detenido de todas las variables x\$(i)
coma	Detecta si pulsamos el punto decimal
w	Controla RETURN o ENTER
chs	Detecta si cambiamos de signo al exponente
x	Almacena n.º del registro 'x'
y	Almacena n.º del registro 'y'
e	Controla el exponente
m\$(a)	Almacena registro 'x'
xe	Almacena la suma estadística de los registros 'x'
xe²	Almacena la suma estadística de los cuadrados de los registros 'x'
x media	Obtiene la media de xe
ye	Almacena la suma estadística de los registros 'y'
ye²	Almacena la suma estadística de los cuadrados de los registros 'y'
y media	Obtiene la media de ye
re	Contador de registros estadísticos
xy	Almacena la suma del producto de los registros 'x' e 'y'
Sx	Almacena la desviación estándar de los registros 'x'
Sy	Almacena la desviación estándar de los registros 'y'

Análisis

POSICIONES DE PANTALLA

A la hora de escribir mensajes en pantalla y realizar distintas operaciones con ella, es muy importante saber qué coordenadas tiene cada posición de carácter en ella. Análisis con posiciones de pantalla, pretende facilitar la comprensión de éstas con un claro ejemplo gráfico, que dibuja las coordenadas y la posición en pantalla. El movimiento del carácter representado, se realiza por medio de las teclas de cursor.

10,20 Mensajes habituales.

30 Limpia la basura de la pantalla.

40 Cambia el color del borde de la pantalla a azul pastel.

60 Define una ventana en la parte inferior de la pantalla, de una extensión de 21 caracteres.

70 Fija el color del fondo de la ventana en rojo brillante, el equivalente a la tinta núm. 3, y el de la escritura en la misma, en azul brillante, tinta núm. 2.

80 Provoca que se haga visible el color de la ventana.

100 Establece las coordenadas iniciales de X e Y.

110 Salto a la línea 180, que hace que aparezca en la ventana el texto Cor. X 20 Cor. Y 12.

120-160 Controla los límites máximos y mínimos de las coordenadas de pantalla, de forma que las líneas 240-270 no produzcan valores fuera de la misma.

180 Escribe en la ventana, el mensaje Cor. X Cor. Y, colocando en su lugar correspondiente los valores de X e Y.

190 Crea las variables Z y W, dándoles el mismo valor que a X e Y, para utilizarlas convenientemente, en el borrado de la flecha que marca la posición en pantalla.

210 Imprime en la posición de coordenadas X e Y, el carácter cuyo código es el 242, es decir, la flecha apuntando hacia la izquierda.

220 Un ciclo de retardo para evitar el parpadeo.

240 Comprueba si se ha pulsado la tecla de cursor hacia arriba, en cuyo caso decrementa la coordenada y en una unidad.

250 Realiza la misma operación, ahora con la tecla de cursor hacia abajo, incrementando en caso de que ésta sea pulsada la coordenada y en una unidad.

260 Si se ha pulsado la tecla de cursor hacia la derecha, incrementa en uno la coordenada X.

270 Decrementa en uno la coordenada X, cuando se pulsa la tecla de cursor hacia la izquierda.

280 Efectúa el borrado de la flecha en pantalla. Para lo cual comprueba si las coordenadas X e Y se han alterado, comparándolas con Z y W, en cuyo caso realiza el borrado de la posición anterior de la flecha, procediendo después con la llamada a la línea 130 a la impresión de las nuevas coordenadas en la ventana y la flecha en su nueva posición.

290 Evita que se detenga el programa, volviendo a chequear el teclado, en busca de una tecla de cursor pulsada.



P ara que tus dedos no realicen el trabajo duro, M.H. AMSTRAD lo hace por ti. Todos los listados que incluyen este logotipo se encuentran a tu disposición en un cassette mensual, solicítalos.

Es usted capaz de tomar el relevo del General Montgomery...

& Juegos ESTRATEGIA

le presenta en exclusiva
el WAR GAME, para Spectrum,
de mayor éxito en Inglaterra:

ARNHEM

(operación «Market Garden», basada
en un hecho real de la Segunda Guerra
Mundial)

Si no lo encontrara en su kiosco puede solicitarlo direc-
tamente a nuestra editorial sin
gastos de envío alguno por
su parte. No demore
su pedido, hay un
número limitado
de cassettes.

Ya está a la venta!

Y ahora también para
AMSTRAD
Compatible todos los modelos



Recorte o copie este cupón y envíelo a Hobby Press, S. A. Apdo. de Correos 54.062. Madrid.
Deseo recibir en mi domicilio, sin gastos de envío alguno por mi parte, la cinta ARNHEM,
al precio de 995 pesetas.

Nombre _____ Edad _____
Dirección _____
Localidad _____
Código _____
La forma de pago elegida es la que señalo con una cruz.
☐ Giro Postal n.º _____
☐ Tarjeta Visa n.º _____
Press, S.A. ☐ Talón nominativo a Hobby
Fecha de caducidad de la tarjeta _____
Fecha y Firma: _____

Provincia _____ Teléfono _____

HOBBY PRESS, S.A. Editamos para gente inquieta.

ARNHEM

Los War Game, son una rara especie de programas, que las casas de software y los usuarios de ordenadores, centrados en presentar juegos en los cuales el uso del Joystick es obligado, tienen un tanto abandonados.

Nos referimos a ese tipo de programas en los que el uso de la inteligencia y la planificación a largo plazo, con una serie de recursos limitados, ocupan un lugar predominante.

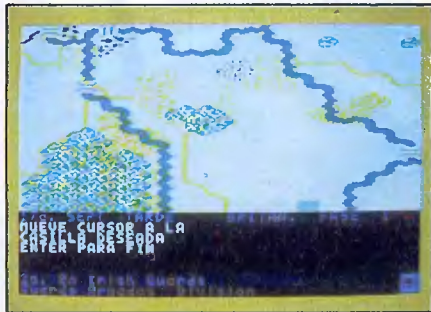
U

n War Game, tiene por objeto reproducir lo más exactamente posible, aquellas batallas cuyo desenlace ha sido de importancia capital para el transcurso de la Historia.

Antes de llegar al mundo del ordenador, los War Game eran juegos de mesa, en los cuales era imprescindible la presencia de dos jugadores, y que debido a las dimensiones de los mapas y la gran cantidad de fichas que deben manejarse, obligaban a los amantes de estos juegos a realizar en un solo día una maratónica partida de seis u ocho horas.

Con la llegada de estos programas al mundo del ordenador, tenemos la posibilidad de elegir entre jugar contra un amigo o desafiar al ordenador, faceta muy interesante porque no siempre encontramos a alguien que sea entusiasta de estos juegos.

También evitamos el problema de tener que jugar la partida de un tirón, porque basta con grabarla y proseguir el juego otro día. Pero la



Compatible: CPC/464, CPC/664 y CPC/6128

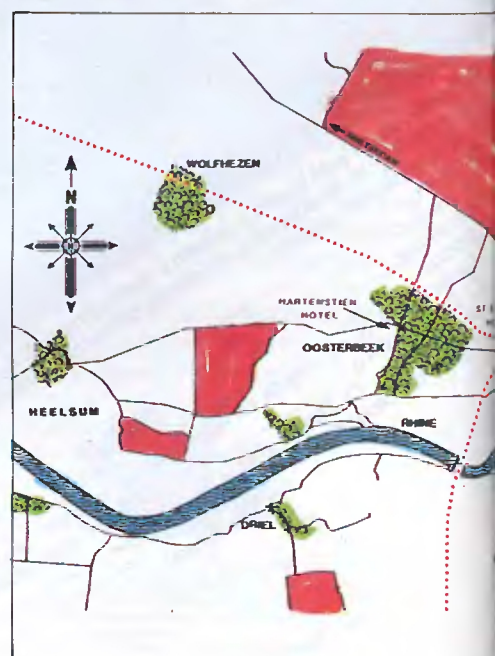
ventaja más grande que aporta el ordenador a estos juegos, es que además de los clásicos factores de combate y de movimiento, propios de cada unidad. En el comportamiento en combate de las mismas, influyen otros detalles mucho más sofisticados.

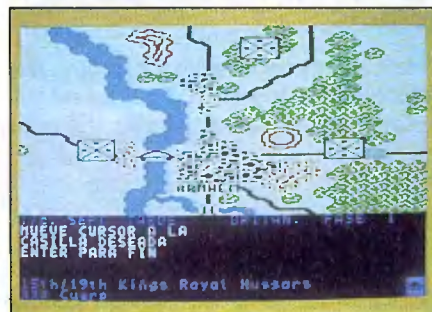
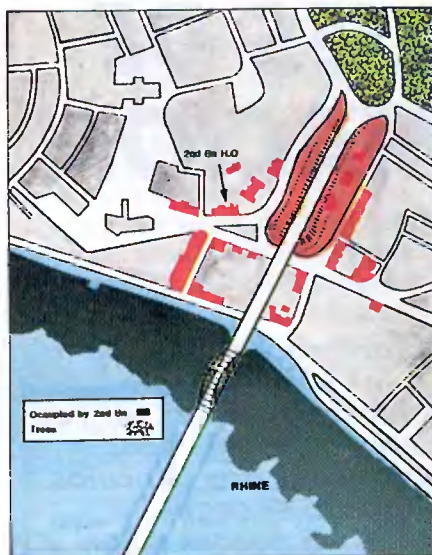
Los nuevos factores son:

La moral de las tropas, muy importante en las batallas de verdad y las bajas que sufre una unidad en combate.

En los juegos de mesa, estas facetas eran inexistentes y cuando una determinada unidad combatía; o era aniquilada; o seguía intacta con todo su poderío y capacidad de lucha, cosa que no se ajusta a la realidad.

Gracias a los ordenadores, se ha conseguido en este tipo de juegos una aproximación mucho más grande a una batalla de verdad.





Al enfrentarnos con uno de estos War Games, debemos utilizar al máximo nuestros recursos intelectuales, la moral de las tropas, capacidad de combate, terreno sobre el que nos encontramos, posición de las unidades enemigas más cercanas, factor de combate de las mismas, etc., son algunos de los detalles que hemos de considerar a la hora de entrar en combate.

La batalla del Arnhem, tuvo lugar en el mes de septiembre de 1944, en ella los aliados pretendían flanquear la línea defensiva alemana Sigfrido por Holanda y de este modo ocupar el corazón de la zona industrial alemana del Rhur.

Ante este marco histórico, fatal para el destino de Alemania, tomamos el mando de las fuerzas nazis y nos preparamos para la defensa de nuestras líneas.

Mr. Joystick

En la confección del programa se han estudiado detenidamente la composición de los ejércitos y el poderío de las distintas unidades, de manera que al principio de la batalla, los dos oponentes tengan los mismos efectivos.

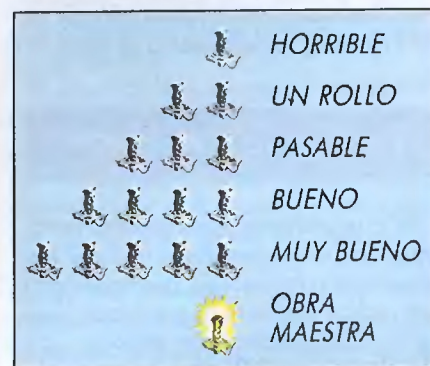
Para ello, aunque las unidades sean totalmente diferentes, los factores de combate y movimiento se han calculado de forma que los de un ejército equilibren a los de otro.

La victoria está en nuestras manos, solamente hemos de dirigir las operaciones desde el estado mayor de nuestros ejércitos, en la pantalla de nuestro ordenador, donde los distintos informes de movimiento de tropas y zonas de batalla, llegarán puntualmente.

Solamente nuestra habilidad táctica y el sentido de la estrategia nos llevarán al triunfo.

El juego se realiza de manera enteramente gráfica, sobre el mapa de la zona de combate y no requiere la introducción de ninguna línea de texto, lo que simplifica extraordinariamente nuestra labor, permitiéndonos concentrarnos al máximo en el transcurso de la batalla.

Los amantes de los War Games, encontrarán la horma de su zapato, los que no hayan tenido la posibilidad de enfrentarse a uno de estos juegos, tendrán oportunidad de utilizar su astucia e inteligencia para vencer a las fuerzas comandadas por el ordenador.



LENGUAJES DE PROGRAMACION (Y III)

A lo largo de este artículo, tercero y último de la serie de introducción a los lenguajes de programación, vamos a tratar de analizar tal vez la parte más apasionante de los mismos. Los lectores que hayan seguido los dos artículos anteriores recordarán que en el segundo establecimos una clasificación, un poco a priori y harto discutible, de los lenguajes por «categorías de aplicación».

Nos quedaba por hablar de dos de ellas: los lenguajes dedicados al aprendizaje y la computación «**simbólica**». Como ambas están íntimamente relacionadas, podemos comenzar por la que más nos guste; elegimos la segunda.

A primera vista, el hecho de que los ordenadores procesen números parece algo completamente normal y no hacen más que responder a una idea de diseño que pretende resolver la mayoría de los problemas computables.

En esto de la **mayoría** está la clave, porque hay una gran cantidad de tareas que revisitan cada vez más importancia e implican tratar con símbolos más que con números, con palabras y reglas de conocimiento más que con cifras.

Los lenguajes existentes para los computadores no estaban muy preparados que dijéramos para tratar estos temas, hasta que en 1960 mister John McCarthy desarrolló LISP (de LISt Processing, procesamiento de listas).

La llegada de LISP

LISP es un intérprete, pero presenta unas características revolucionarias en cuanto al tratamiento de la información, que hasta cierto punto echan por tierra la arquitectura Von Neuman de los ordenadores clásicos.

En efecto, para LISP todo son listas de objetos y no existe diferencia alguna entre programa y datos, cosa que el resto de los lenguajes distinguen claramente, por hipótesis podríamos decir.

Esto implica que cadenas de caracteres, nú-

meros, estructuras de datos y las propias instrucciones de intérprete LISP pueden ser tratados y manipulados por el propio LISP de la misma forma, pues cada lista posee un significado simbólico, que depende exclusivamente del contexto en que la «expresión simbólica» se encuentre.

Además, LISP, como muchos lenguajes modernos, es un idioma orientado a funciones (*procedural*), de modo que nuevas extensiones del LISP están escritas en LISP.

Por otra parte, McCarthy, cuando diseñó el lenguaje, trató de adecuarlo a la resolución de problemas que interesan a la Inteligencia Artificial, más que nada, por lo que implementó en LISP una particularidad que hasta entonces nunca se había visto antes en tan gran escala: la recursividad.

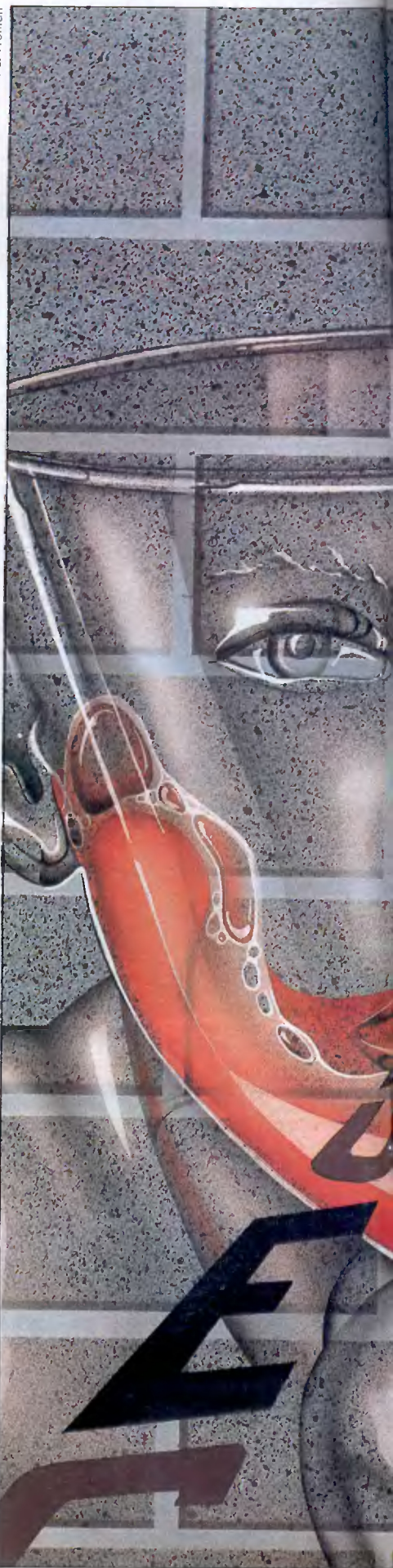
Recursividad e Inteligencia

Esto significa que una función LISP, que requiera una serie de parámetros, puede llamarse a sí misma una y otra vez mientras en el cuerpo de la función los parámetros se alteran, consiguiendo alcanzar al final una condición límite en la cual la recursión termina y el problema se resuelve «sumando» los resultados de cada llamada que la función se ha hecho a sí misma.

Esto, que suena rarísimo, realmente es una forma muy humana de resolver un problema, esto es, describiéndolo en términos de sí mismo, pero con un grado de complejidad menor, hasta que dicha complejidad sea nula.

Un ejemplo. Supongamos que yo tengo la lista (a b c d), y quiero hallar su longitud, o sea, el número de objetos que la integran. La manera más simple de resolver mi problema es decirle al ordenador que los vaya contando de uno en uno y sumando 1 a ese contador hasta que el fin de la lista se encuentre.

F. L. Frontón





Para conseguir esto, yo puedo decir:

- 1) La longitud de la lista, sin ningún elemento, esto es, vacía, es obviamente 0.
- 2) Sea cual sea la longitud de la lista, siempre será cierto que valdrá la suma de la longitud del primer elemento, o sea, 1, más la longitud del resto de la lista.

Aquí aparece la recursividad, porque estoy describiendo la longitud de una lista en función de la longitud de una lista, pero con un elemento menos. La condición límite que pone fin a la recursión es la uno, porque al ir dividiendo la lista sucesivamente en dos trozos una y otra vez, tarde o temprano alcanzará la lista vacía.

En LISP, el programa sería el siguiente:

```
(defun long (lista)
  (cond ((null lista) 0)
        (t (add 1 (long (cdr lista)))))
```

Una llamada a esta función de la forma:

```
(long (a b c d))
```

dará como resultado 4.

Programas no jerárquicos

Obsérvese que esta manera de construir programas permite crear estructuras de datos **ramificadas** más que jerárquicas.

Por ejemplo, en PASCAL suele haber un procedimiento principal que maneja y llama a los demás en el momento adecuado, dando al programa final una estructura claramente jerárquica.

LISP proporciona una flexibilidad mucho mayor, porque cada función puede llamarse a sí misma o a otras, que a su vez se llaman a sí mismas, desde cualquier parte del programa. Yo podría escribir tres funciones llamadas f1, f2 y f3, por ejemplo, y hacer que cada una entrara en la definición de la otra:

```
(defun f1 (parámetros1)
  (haz-cosas)
  (f2 (haz-cosas)))

(defun f2 (parámetros2)
  (f1 (haz-más-cosas))
  (f3 (parámetros3)))

(defun f3 (parámetros3)
  (f1 (f2 (parámetros 3))))
```

Si las condiciones de salida de recursividad se eligen adecuadamente, este programa funcionará y no puede decirse que sea muy jerárquico, más bien parece bastante «arbóreo». Un follón, vamos.

Este lío permitió resolver una serie de problemas inabordables hasta ese momento acerca de cómo tratar el conocimiento, porque, según parece, nuestra mente lo hace de una manera relativamente parecida, salvando las distancias.

Resulta un poco difícil explicar cómo funciona un lenguaje dedicado al tratamiento simbólico de la información, pero espero que se me crea que LISP supuso una auténtica revo-

lución en el campo de la Inteligencia Artificial y de los lenguajes en general, porque permitió a los investigadores materializar y probar sus ideas el contar con la herramienta de trabajo adecuada.

En busca de una sintaxis más sencilla

Los lectores que no estén familiarizados con el lenguaje habrán observado que su sintaxis es muy extraña y sobre todo, está llena de paréntesis por todas partes (*los que conocen LISP, también lo han observado en su propia carne*), no resulta especialmente amigable.

Esto y otro asunto muy importante del que vamos a hablar ahora mismo, hizo que se desarrollaran multitud de lenguajes-hijo del LISP especializados en manipular información simbólica, pero con una sintaxis y un modo de programar mucho más sencillo (¿menos potente?).

El asunto muy importante resulta ser algo obvio y triste: para que un programa funcione, yo tengo que saber antes que el ordenador exactamente, cómo va ha hacerlo, es decir, debo conocer por anticipado el flujo del programa, las variables, constantes y estructuras de datos necesarias para resolver mi problema.

Por esto, a la mayoría de los lenguajes se les denomina «**imperativos**» (*LISP es medio imperativo*), porque me limito a ordenar a la máquina paso a paso lo que debe hacer y cómo.

Resulta que soy YO el que hace el trabajo del ordenador, y encima me creo que lo estoy programando.

Los lenguajes declarativos

Algún tipo listo pensó que no estaría de más que el ordenador trabajara un poquito; al fin y al cabo, a mí personalmente me importa un bledo todo lo relativo al mecanismo computacional concreto de una máquina. Me da igual cómo se resuelva mi problema, siempre que se resuelva.

En este sentido, analicemos lo que el programador hace normalmente cuando crea un programa. Primero, se describe a sí mismo el problema en función básicamente de 2 cosas: unos hechos, de los que se parte y unas reglas, que los relacionan. Así, en un programa Basic, por ejemplo, yo parto de la incontrovertible realidad de que la variable X vale cero, más que nada porque la pongo yo y luego, en alguna parte perdida del programa, voy y digo:

```
IF X=0 THEN haz-tal
```

El problema empieza en que si la variable X se altera a lo largo del programa por lo que sea, yo tengo que saberlo a la hora de escribir la sentencia condicional, es decir, debo co-

nocer perfectamente cómo se computa la variable X; estoy metido hasta el cuello en un lenguaje imperativo.

Para evitar esto, debe existir un lenguaje que le baste recibir una descripción del problema en términos de hechos y reglas y, partiendo de ellos, realice él solito el trabajo verdaderamente duro. Existe, se llama PROLOG y es la base de la Quinta Generación de Ordenadores.

Programación lógica

No sé si los lectores se percatarán de la revolución que representa el hecho de que la descripción lógica de un programa sea ya el programa, sin más.

Por fin, tenemos que limitarnos sólo a plantear un problema a la máquina en lugar de resolverlo nosotros pasito a pasito, y dejarle al ordenador sólo los cálculos matemáticos pertinentes.

Para tratar de comprender un poco esto, vamos a escribir en PROLOG el programa que permita hallar la longitud de una lista. Primero de todo, vamos a describir el problema en castellano:

- 1) La lista vacía tiene una longitud 0.
- 2) La lista no vacía tiene una longitud Z si podemos dividirla en dos partes, un miembro y el resto, y el resto tiene una longitud Y tal que sumada a 1 da Z.

Léase esto con atención y, cuando por fin no se entienda, mírese atentamente lo siguiente:

La lista (a b c d) puede dividirse en dos: (a) y (b c d).

Podemos decir que la longitud de toda la lista es 4 porque la longitud de la segunda parte es 3 y 3 más 1 es igual a 4.

¿A que resulta obvio?

Ahora en PROLOG:

() longitud-es 0

(xIX) longitud-es Z IF X longitud-es y AND SUM (Y 1 Z).

En donde () representa la lista vacía e IF, AND y SUM son primitivas del PROLOG, mientras que (xIX) es la forma de decirle al ordenador que nos habemos con una lista dividida en su primer miembro y el resto.

Insisto en el detalle de que nosotros en ningún momento le hemos indicado al ordenador el mecanismo exacto para hallar la longitud de una lista. Nos hemos limitado a describirle a la máquina el significado del concepto longitud_de_una_lista usando el sentido común. Obsérvese también que hemos empleado, encima, la forma más cómoda, esto es, describir longitud_de_una_lista en términos de longitud_de_una_lista. Recursividad al canto.

En BASIC, PASCAL o cualquier otro lenguaje imperativo usted tendría que saber cómo se halla exactamente la longitud de una lista **computadoramente** hablando.

Bien, espero que quede claro porque se dice que PROLOG es un lenguaje **declarativo**, y porque se usa en Inteligencia Artificial para la construcción de sistemas expertos basados en reglas lógicas, en la descripción de situaciones.

La Quinta Generación

Los lenguajes declarativos tienen, por ahora, una serie de limitaciones, fundamentalmente que son muy lentos y consumen gran cantidad de memoria.

Sin embargo, estos puntos negros de PROLOG y otros no son completamente achacables al lenguaje en sí. El problema estriba en que todavía no disponemos de un hardware adecuado para los lenguajes declarativos.

PROLOG posee una característica que, en palabras técnicas, se llama **transparencia referencial**, y se puede definir diciendo: **un lenguaje es referencialmente transparente si prohíbe asignar a una misma variable diferentes valores durante la ejecución del programa.**

Esto, en palabras sencillas, indica que una parte de un programa PROLOG o de cualquier lenguaje declarativo es completamente independiente de las otras y, por tanto, una aplicación declarativa puede ser procesada en paralelo, computando a la vez la mayor cantidad de trozos posible del programa.

Esta idea del procesamiento paralelo, que implica que un ordenador posee más de un microprocesador trabajando conjuntamente con otros en la resolución de un problema, es la base, junto con el PROLOG, de la Quinta Generación de Ordenadores, el famoso proyecto japonés.

Japón pretende dotarnos, para los años 90 más o menos, de ordenadores que se programen en gran parte a sí mismos, a los que podamos dirigirnos mediante la palabra hablada e incorporen, como diseño de base, una serie de sistemas expertos capaces de entenderse con bases de conocimientos.

Los problemas son de muy difícil solución, pues en realidad se trata de dotar de inteligencia a las máquinas, o al menos de que las simulen tan convincentemente que no seamos capaces de distinguirlas de la humana.

Por razones económicas y políticas, además de operacionales, PROLOG ha sido escogido para ser el lenguaje de los ordenadores pensantes de finales del siglo XX. Hemos dado un gran paso desde los tiempos del ENIAC, las máquinas regidas por válvulas y el uso exclusivo del código máquina. Sólo la tecnología física nos separa de crear la primera forma de vida inteligente conocida aparte del hombre mismo.

Los lenguajes que lo harán posible ya existen, y el mayor desafío tecnológico y filosófico del siglo está a punto de nacer y explotar en nuestras vidas.



PLACON

Placon es un programa de compatibilidad completamente adaptado al Plan Contable Español y de lo más potente que hemos visto.

Como está dirigido a un sector muy concreto de usuarios de Amstrad, hemos pensado que lo mejor es pasar a la descripción detallada del programa sin más preámbulos.

Únicamente una aclaración antes de continuar: Placon requiere para poder funcionar dos unidades de disco. Incluso está prevista su utilización con disco rígido.

Las funciones del programa están numeradas y se ha mantenido esta clasificación para ser coherentes con la estructura del mismo.

OPCION 10: Mantenimiento del Plan de Cuentas

A. OBJETO DE LA OPCION

Esta opción permite dar de alta, modificar, dar de baja y listar las cuentas.

B. CUANDO SE UTILIZA

Cuando se necesite.

C. COMO SE UTILIZA

Se llega a ella desde el menú principal pulsando 10 y ENTER (o bien situándose con el cursor y ENTER).

Tiene dos formas de elegir opciones:

— Con las teclas de flechas hacia arriba y hacia abajo.

— Pulsando el número elegido.

A continuación de ambas opciones hay que teclear ENTER.

D. TERMINAR

Con la opción 0 se vuelve al menú principal.

OPCION 10 — 1: Altas

A. OBJETO DE LA OPCION:

Introducir cuentas en el maestro, tanto cuentas de agrupación como de detalle.

B. CUANDO SE UTILIZA

Cuando se quiera introducir una cuenta.

C. COMO SE UTILIZA

Eligiendo la opción 10 (MANTENIMIENTO DE FICHEROS DEL PLAN DE CUENTAS) en el menú principal y luego la opción 1 (ALTAS) en el menú secundario.

Al teclear el número de cuenta, observe que se admiten caracteres alfanuméricos y el signo— (signo menos).

D. TERMINAR

Para TERMINAR teclee escape a número de cuenta.

Las entradas de cuentas se graban en un fichero auxiliar que luego se clasifica y mezcla con el del plan contable.

OPCION 10 — 2: Modificar

A. OBJETO DE LA OPCION

Modificar:

— La descripción de las cuentas.

— La asignación a masa patrimonial.

— La asignación a activo o a pasivo.

B. CUANDO SE UTILIZA

Cuando se desee realizar lo anteriormente señalado.

C. COMO SE UTILIZA

Teclee 10 en el menú principal y 2 en el secundario.

Se le preguntará por el código de la cuenta a modificar.

Teclee la cuenta, el sistema la buscará y le mostrará en pantalla los valores actuales para que proceda a su modificación.

Si no la encuentra, le mostrará la más próxima.

D. PARA TERMINAR

Teclee ESCAPE en cualquier momento.

OPCION 10 — 3: Bajas

A. OBJETO DE LA OPCION

Anular cuentas.

B. CUANDO SE UTILIZA:

Puede utilizarse en cualquier momento para anular cuentas que no tengan ni saldo ni movimiento.

Normalmente después de cerrar el ejercicio.

C. COMO SE UTILIZA

Se le preguntará por el código de cuenta a anular y si no la encuentra tomará la más cercana, a continuación le pedirá conformidad para anularla definitivamente.

OPCION 10 — 4: Listado

A. OBJETO DE LA OPCION

Tener un listado del plan de cuentas, tanto por pantalla, como por impresora.

B. CUANDO SE UTILIZA

Cuando se necesite.

C. COMO SE UTILIZA

Tiene opción a seleccionar en qué cuenta empieza y termina el listado y el periférico (pantalla o impresora) de salida.

OPCION 11: Mantenimiento del Fichero de Masas Patrimoniales

A. OBJETO DE LA OPCION

Las masas patrimoniales se usan para agrupar cuentas de mayor, según el grado de liquidez, al imprimir el balance de situación.

B. CUANDO SE UTILIZA:

— Cuando se necesite, normalmente al iniciar la aplicación.

C. COMO SE UTILIZA:

Dispone de tres opciones:

— Si teclea un número entre 1 y 20 el cursor se posicionará en la masa referenciada y lo que teclee será el nuevo contenido.

— Si teclea (L) se imprimirá un listado de las masas.

— Si teclea (0) se grabará el fichero de masas con los nuevos contenidos y el programa volverá al menú anterior.

— Si teclea un número mayor de 20 volverá a escribir la pantalla.

D. PARA TERMINAR

— Si teclea (0) se grabará el fichero de masas con los nuevos contenidos y el programa volverá al menú anterior.

OPCION 12: Mantenimiento del Fichero de Conceptos

A. OBJETO DE LA OPCION

El fichero de conceptos contiene 45 descripciones de 20 caracteres cada una que utilizaremos para facilitar y ahorrar espacio al introducir los asientos.

B. CUANDO SE UTILIZA

— Cuando se necesite, normalmente al iniciar la aplicación.

C. COMO SE UTILIZA

Dispone de tres opciones.

— Si teclea un número entre 1 y 45 el cursor se posicionará en el concepto referenciado y lo que teclee será el nuevo contenido.

— Si teclea (L) se imprimirá un listado de los conceptos.

— Si teclea (0) se grabará el fichero de conceptos con los nuevos contenidos y el programa volverá al menú anterior.

— Si teclea un número mayor de 45 volverá a escribir la pantalla.

D. PARA TERMINAR

— Teclee (0), se grabará el fichero de conceptos con los nuevos contenidos y el programa volverá al menú anterior.

OPCION 20: Anotación de Asientos

A. OBJETO DE LA OPCION

Anotar los movimientos de la contabilidad.

B. CUANDO SE UTILIZA

En principio cuando se quiera con las siguientes restricciones:

1. Si es el primer asiento después del de apertura asegúrese de haber realizado el proceso de fin de apertura.

2. Si es el primer asiento del mes, asegúrese de haber realizado el proceso fin de mes.

3. En cualquier caso es necesario haber realizado la inicialización del disco.

C. COMO SE UTILIZA

La entrada de datos es del tipo de filas y columnas, además de las teclas especiales de anotación de asientos: (VER SU ORDENADOR, PUES SON DISTINTAS PARA CADA MODELO.)

OPCION 30: Modificación de Apuntes

A. OBJETO DE LA OPCION

Esta opción permite la modificación de cualquier campo, de cualquier apunte, de cualquier asiento ya actualizado.

B. CUANDO SE UTILIZA

— Cuando se necesite.

C. COMO SE UTILIZA

Seleccione MODIFICACION DE APUNTES (opción 30) en el menú principal.

Tiene tres opciones:

1. Pulsando un número de referencia aparece el apunte correspondiente.

Pulsando un número del 1 al 8, el cursor se posiciona en el campo correspondiente y una vez en ese campo se puede modificar.

En el campo 1 con poner una A (anulado) o una N (no anulado) el programa escribe lo demás.

D. TERMINAR

Se sale pulsando una F en la primera pantalla de la opción.

E. OBSERVACIONES

No se le permitirá modificar apuntes que no existen.

OPCION 31: Incorporación de Asientos Externos

A. OBJETO DE LA OPCION

Esta opción lee ficheros realizados por otras aplicaciones (Nómina, Facturación —ALFA—, e incluso Contabilidad General) y los incorpora a la contabilidad en curso.

El proceso es parecido al de introducción de asientos. La diferencia es que, en vez de leer el teclado, se leen los datos del fichero grabado por otra aplicación.

Los apuntes se incorporan al fichero auxiliar de entrada de apuntes. Puede modificarlos con la opción de anotación de asientos exactamente igual que si lo hubiera tecleado. Hasta que no lo actualice no se considera en los listados.

B. CUANDO SE UTILIZA

Se puede utilizar cuando se quiera.

Puede serle de mucha utilidad si usted tiene un negocio con varias sucursales, tiendas o departamentos, así usted puede llevar una con-



tabilidad para cada departamento y luego agrupar todas en una general sin tener que teclear nada.

C. COMO SE UTILIZA

Si selecciona esta opción, primero le pregunta por el nombre del fichero.

Luego le pregunta por la estructura de los ficheros:

— Si usted va a incorporar un fichero del Placon o Alfa (almacén y facturación) teclee ENTER a todos los datos de estructura, pues tomará por defecto los valores de este fichero.

— Si va a incorporar otro fichero teclee los datos de su estructura.

OPCION 40: Fin de Apertura

A. OBJETO DE LA OPCION

Actualizar el campo del saldo inicio de ejercicio.

B. CUANDO SE UTILIZA:

Después de grabar el asiento de apertura y antes de realizar otro asiento.

C. COMO SE UTILIZA

Lo primero que se le pedirá es que teclee la clave de seguridad.

La clave de seguridad salvo que se solicite lo contrario es TOMA. Al teclearla no se ve lo que se teclea en la pantalla. Es preciso teclear mayúsculas. Si no se teclea exactamente, el programa vuelve al menú anterior.

D. PARA TERMINAR

Al terminar el proceso vuelve directamente al menú anterior. Para no realizar el proceso basta con teclear erróneamente la clave de seguridad.

E. OBSERVACIONES

El uso inadecuado de esta opción desfigura la cuenta de explotación (en la variación de existencias) y el balance de sumas y saldos (en el saldo del ejercicio anterior).

OPCION 41: Fin de Mes

A. OBJETO DE LA OPCION

Actualizar los campos sumas debe mes anterior y sumas haber mes anterior.

B. CUANDO SE UTILIZA

Después de terminado un mes y de listar todos los informes necesarios y antes de introducir los asientos del nuevo mes.

C. COMO SE UTILIZA

Lo primero que se le pedirá es que teclee la clave de seguridad.

D. PARA TERMINAR

Al terminar el proceso vuelve directamente al menú anterior.

Para no realizar el proceso basta con teclear erróneamente la clave de seguridad.

OPCION 42: Fin de Periodo

A. OBJETO DE LA OPCION

Es actualizar los campos de sumas del debe y sumas del haber del periodo anterior e inicializar el fichero de asientos.

No es prudente realizar este proceso si no hay dos copias del disco de datos (se guardarán como históricos).

B. CUANDO SE UTILIZA

Este proceso lo ha de realizar cuando se lleve el fichero de movimientos.

C. COMO SE UTILIZA

Lo primero que se le preguntará es la clave de seguridad. Si la teclea mal volverá al menú.

A continuación comienza la actualización de las cuentas (que podrá ver en pantalla), y la inicialización del fichero de asientos.

D. PARA TERMINAR

Vuelve automáticamente al menú.

OPCION 43: Regularización de Cuentas de Gestión

A. OBJETO DE LA OPCION

Generar un asiento cuyos apuntes saldan las cuentas de gestión.

B. CUANDO SE UTILIZA

Después de:

— Introducir todos los apuntes del ejercicio.

— Listar la cuenta de explotación.

— Antes de realizar el proceso de fin de ejercicio.

C. COMO SE UTILIZA

Lo primero que se le preguntará es la clave de seguridad. Si la teclea mal volverá al menú.

Después le preguntará el número de documento y la fecha del asiento de regularización. A continuación el programa escribe el asiento de regularización.

D. PARA TERMINAR

Vuelve automáticamente al menú.

OPCION 44: Fin de Ejercicio

A. OBJETO DE LA OPCION

Poner a cero los acumulados del debe y del haber de todas las cuentas y generar el asiento de apertura (que queda en el fichero auxiliar de entrada de apuntes) también inicializa el fichero de asientos.

B. CUANDO SE UTILIZA

Al terminar el ejercicio contable.

C. COMO SE UTILIZA

Lo primero que se le preguntará es la clave de seguridad. Si la teclea mal volverá al menú.

Y comienza la actualización de cuentas e inicialización del fichero de asientos.

D. PARA TERMINAR

Vuelve automáticamente al menú.

OPCION 50: Listado del Diario

A. OBJETO DE LA OPCION

Listar el diario.

B. CUANDO SE UTILIZA

— Cuando se necesite.

C. COMO SE UTILIZA

Tiene opción a:

— Definir un intervalo de fechas.

— Definir un intervalo de números de asiento.

— Definir un intervalo de referencias.

— Que se impriman sólo las cuentas que empiezan por la serie de dígitos. (P.e. Si tecla 7 se imprimirán sólo los apuntes que se han hecho a cuentas del grupo 7, si tecla 570 se imprimirán los que hacen referencia a la Caja, etc.)

— Dar unas sumas de debe y haber iniciales.

— Imprimir o no la descripción de las cuentas.

— Numerar a partir de una página dada (si tiene por ejemplo, listados enero y febrero y le ocupan 26 páginas, teclee 27).

En el listado saldrán los apuntes que cumplan todas las condiciones especificadas.

D. PARA TERMINAR

Al terminar el listado, el programa vuelve automáticamente al menú anterior.

OPCION 51: Listado del Mayor

A. OBJETO DE LA OPCION

Listar el mayor.

B. CUANDO SE UTILIZA

— Cuando se necesite.

C. COMO SE UTILIZA

Tiene opción a elegir:

— El periférico de salida del informe, con la opción de poder utilizar fichas standar.

— El rango de cuentas a listar.

— El intervalo de listado de movimientos (que puede definirlo por fecha inicial y final, número de documento inicial y final o número de referencia inicial o final).

En el listado saldrán los apuntes que cumplan todas las condiciones especificadas.

D. PARA TERMINAR: Al terminar el listado el programa vuelve automáticamente al menú anterior.

OPCION 52: Balance de Sumas y Saldos

A. OBJETO DE LA OPCION

Listar un balance de sumas y saldos.

B. CUANDO SE UTILIZA

— Cuando se necesite.

C. COMO SE UTILIZA

Tiene opción a definir:

— En qué cuenta empieza y termina el listado.

— Los niveles de rotura de totales.

— El modelo de listado (con datos del mes, datos del ejercicio o ambos).



Quando no imprima cuentas de detalle (nivel 7) notará que algunas líneas tardan en salir. Esto es porque los valores de las cuentas de agrupación se calculan de los de las de detalle.

D. PARA TERMINAR

Al terminar el listado el programa vuelve automáticamente al menú anterior.

OPCION 53: Balance de Situación

A. OBJETO DE LA OPCION

Obtener el balance.

B. CUANDO SE UTILIZA

— Cuando se necesita. Para que tenga sentido se debe usar después de introducidos todos los apuntes del mes y un asiento de regularización de la cuenta de existencias (contrapartida en el grupo 8).

El balance de final de año debe obtenerse después de la regularización de existencias y antes del proceso de fin de ejercicio.

C. COMO SE UTILIZA

Este listado no ofrece ninguna opción de selección.

D. PARA TERMINAR

Al terminar el listado le preguntará si desea otra copia. Teclee N para terminar.

OPCION 54: Cuenta de Explotación

A. OBJETO DE LA OPCION

Obtener el informe.

B. CUANDO SE UTILIZA

— Cuando se necesite. Para que tenga sentido se debe usar después de introducidos todos los apuntes del mes y un asiento de regularización de la cuenta de existencias (contrapartida en el grupo 8).

La cuenta de explotación debe obtenerse después de la regularización de existencias y antes de la regularización de compras y ventas.

En este informe se imprime la variación de las existencias, de las cuentas de compras (Grupo 6) y de las ventas (Grupo 7).

OPCION 55: Explotación Analítica

A. OBJETO DE LA OPCION

Obtener el informe.

B. CUANDO SE UTILIZA

— Cuando se necesite. Para que tenga sentido se debe usar después de introducidos to-

dos los apuntes del mes y un asiento de regularización de la cuenta de existencias (contrapartida en el grupo 8).

La cuenta de explotación debe obtenerse después de la regularización de existencias y antes de la regularización de compras y ventas.

Contiene los mismos datos del informe anterior clasificados en tres grupos. Da el porcentaje de los gastos sobre las ventas. Si las ventas son cero no se imprime este porcentaje.

OPCION 60: Previsión de Cobros y Pagos

A. OBJETO DE LA OPCION

Mantenimiento del fichero de previsiones y listado clasificado de previsiones de cobros y pagos.

B. CUANDO SE UTILIZA

Quando se necesite.

C. COMO SE UTILIZA

Al seleccionar la opción aparecerá una pantalla del tipo de filas y columnas.

D. TERMINAR

Al terminar cada una de las opciones se vuelve al menú secundario y así seleccionando la opción 0 volvemos al menú anterior.

OPCION 62: Otras Opciones

REGENERACION DE INDICES

1. Chequea que el plan de cuentas esté cuadrado en datos del período anterior, si no lo está lo cuadra contra la cuenta 9999999.

2. Chequea que el diario esté cuadrado, si no lo está genera un apunte contra la cuenta 9999999 que lo cuadra.

3. Pone a cero el plan de cuentas.

4. Para cada apunte del diario comprueba la existencia de la cuenta correspondiente.

5. Actualiza las cuentas a través de los apuntes.

OPCION 99: Fin Aplicación

Esta opción es el final obligado de todo proceso contable en el que haya habido modificación de ficheros.

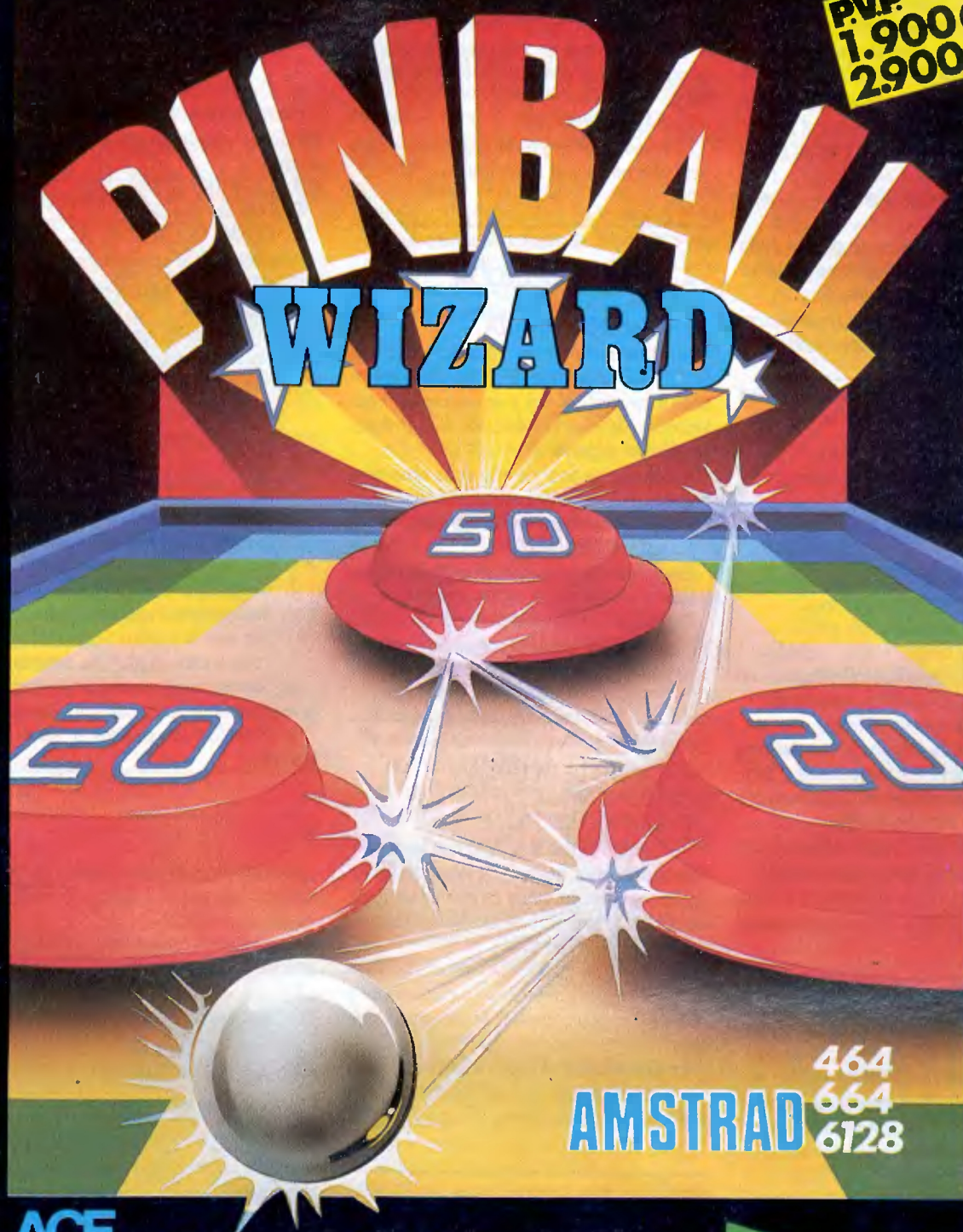
Una vez realizado puede sacar los discos y apagar el ordenador.

CONSULTAS: 51 2547719					
LISTADO		++	PLANCON	++	12-jun-85
M/T Plan de Cuentas	10		Diario		50
M/T Masas Patrimoniales	11		Mayor		51
M/T Conceptos Auxiliares	12		Balance de Sumas y Saldos		52
			Balance de Situación		53
			Cuenta de Explotación		54
			Explotación Analítica		55
M/Tificación Apuntes	30		Previsión de Cobros y Pagos	60	
Asientos Lateral	31		Modificar Instalación	61	
Fin de Operación	40		Regeneración de Indices	62	
Fin de Mes	41				
Fin de Período	42		Fin Aplicación	99	
Regularización Costos	43				
Fin de Ejercicio	44				
					OPCION: 99B:
Cometas	3	Apuntes	2	Mayor	100
					Libres

UN "MILLON" EN JUEGO

Gráficos a color, rapidez de acción y sonido dan a este juego la sensación de una máquina de millón real.

PVP:
1.900 (cinta)
2.900 (disco)



ACE

ACTIVIDADES COMERCIALES ELECTRONICAS, S.A.
Tarragona, 110. Tel. 325 10 58* 08015 Barcelona. Telex 93133 ACEEE

YA DISPONIBLE EN

El Comercio

... Y EN TODAS LAS
TIENDAS ESPECIALIZADAS

3~D VOICE CHESS

Ajedrez tridimensional con voz en castellano
Amstrad CPC 464, CPC 664 y CPC 6128

P.V.P.

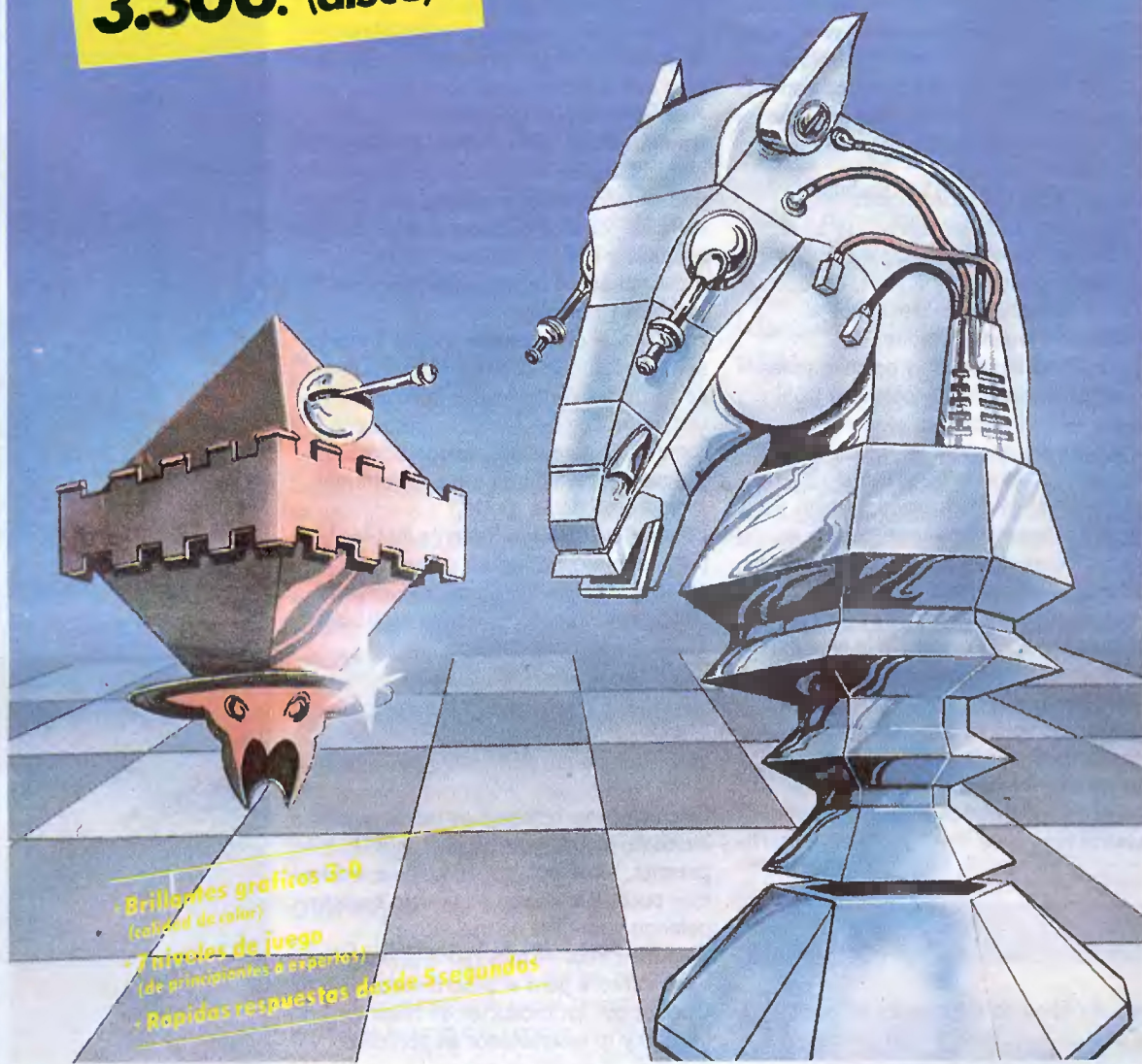
2.300.- (cinta)
3.300.- (disco)



PUBLISHED
DEEP THOUGHT
SOFTWARE

DISTRIBUTED

cp software



Brillantes gráficos 3-D
(colores de color)
7 niveles de juego
(de principiantes a expertos)
Rápidas respuestas desde 5 segundos

Producido en exclusiva para España por:

ACE

Actividades Comerciales Electrónicas, S.A.
Tarragona, 110 · Tel. 325 10 58 · 08015 Barcelona, Télex 93133 AC EE E

YA DISPONIBLE EN



... Y EN TODAS LAS
TIENDAS ESPECIALIZADAS

ENSAMBLADOR DE HISOFT

Algunas personas compran un ordenador para que les solucione, en la medida de lo posible, una serie de problemas o tareas complicadas y tediosas de resolver, normalmente relacionadas con la gestión en el hogar o en el trabajo.

Otros, por el contrario, piensan que han adquirido un interesante juguete además de una herramienta de trabajo, y que merece la pena dedicar algo de su precioso tiempo a averiguar cómo funciona un ordenador, un Amstrad.

E

stas personas están perdidas, porque cuando se quieren dar cuenta el virus de la informática ha penetrado en su cuerpo y acaban convirtiéndose en programadores así, como quien no quiere la cosa.

El primer paso del recién nacido programador suele ser adentrarse en el uso de un lenguaje para poder expresar sus ideas, normalmente el Basic, aunque no sea más que por ser el idioma que el **Amstrad** lleva incorporado.

El **Amstrad Basic** es muy potente: posee un set de comandos muy completo y útil, en casi todos los campos, como ficheros, sonido y gráficos.

No obstante, tarde o temprano el programador ávido alcanza los límites del Basic; simplemente hay cosas que no se pueden hacer, o porque son demasiado lentas o porque el Basic no ha sido diseñado para soportarlas.

Justo en ese punto asoman a la mente las palabras lenguaje máquina. Uno sabe de oídas, más o menos, que el código de máquina no va a causar problemas de velocidad, ni de ocupación de memoria y que permite tomar control absoluto del ordenador, sometiéndolo completamente a nuestros designios.

Código máquina

Pero el código de máquina es un hueso duro de pelar. Cuesta mucho tiempo y esfuerzo aprender a manejarse con él, mucho más que con cualquier lenguaje de alto nivel. Aquí no hay firmware que valga, ni mensajes de error. Un fallo, uno solo y el ordenador se cuelga sin misericordia.

Naturalmente, los comienzos en este terreno se asemejan a una tragedia griega y uno, sabiendo lo que es la desesperación, aprende a valorar las herramientas de programación, cualquier cosa que le facilite un poco el tema.

Aquí aparecen los programas ensambladores, por la puerta grande. Un ensamblador no es más que un programa que pretende salvar el abismo que existe entre los números binarios, el puro código máquina y nuestra cabecita, mucho más acostumbrada a regirse en términos de palabras y símbolos.

Para ello, el ensamblador usa los llamados mnemónicos, que no son más que grupos de letras que corresponden cada uno a una instrucción del Z80 y nos recuerdan, como su nombre indica, la operación que se supone que realizan y a qué parte del microprocesador le afecta.

Esto nos hace la vida bastante más fácil, porque yo escribo texto y el programa ensamblador se encarga de traducirlo a código binario puro, la lengua nativa del Z80 y la única que entiende.

GENA3

Ensambladores hay muchos (no en nuestro mercado, por desgracia), unos más potentes, otros más modestos. Pero, existen muy pocos que reúnen en un solo conjunto potencia y sencillez de uso.

Dentro de estos escasos representantes del buen software para el programador, creemos que los productos de Hisoft en general y su ensamblador en particular, merecen un puesto de honor.

El **GENA3**, ensamblador de Hisoft, es el mejor que hemos encontrado sin la menor duda; ya lo conocíamos porque existe una implementación para el Spectrum, pero la del **Amstrad** suma a las cualidades de la

F. L. Frontón

anterior, toda la potencia del disco integrado, lo cual no es para despreciarlo, ni mucho menos.

Vamos a ver con detalle qué tipo de ensamblador es GENA3 y cómo funciona. GENA3 consta básicamente de dos partes: un editor de pantalla, soberbiamente diseñado, que nos permite introducir el texto en mnemónicos siguiendo unas sencillas reglas de **sintaxis** (*luego lo estudiaremos*), y el propio programa ensamblador que traducirá nuestro texto a números binarios. Como supondrán los lectores que hayan seguido de cerca nuestro curso de código máquina, el ensamblador maneja tres tipos de **objetos**: mnemónicos, etiquetas y expresiones, algunas de bastante complejidad, que involucran cálculos aritméticos y lógicos.

Descripción de GENA3

Además, el ensamblador posee una serie de **directivos del ensamblador**, que así se llaman y son órdenes especiales que producen efectos concretos cuando GENA3 se los encuentra en un fichero de texto creado con el editor de pantalla. Son los siguientes:

Directivos del Ensamblador

Ciertas instrucciones pseudonemónicas son reconocidas por GENA3. Estos directivos no tienen efecto en el procesador Z80 al ejecutar el programa, es decir, no se traducen en código objeto, simplemente dirigen al ensamblador mientras ensambla. Las acciones que estas órdenes mandan efectuar al ensamblador cambian de alguna manera el código objeto generado por éste. Los pseudonemónicos se ensamblan exactamente igual que las instrucciones ejecutables; pueden ser precedidas por una etiqueta necesaria para EQU y seguidas por un comentario. Los directivos existentes son:

ORG expresión

Coloca el puntero en el valor dado por «expresión». Si no eliges la opción 2 ni la 16 y ORG provocaría que el programa invadiera la zona asignada a GENA3, al archivo de texto o a la tabla de símbolos, el mensaje **Bad Org!** aparecerá en pantalla y el ensamblado parará.

EQU expresión

Debe ir precedido por una etiqueta. Otorga a la etiqueta el valor de

Código máquina

«**expresión**». Esta expresión no puede contener ningún símbolo que no haya sido definido previamente (*o si no, obtendremos el *ERROR*13*).

DEFB expresión, expresión...

Cada expresión no puede tener más de 8 bits; el byte correspondiente al valor del puntero recibe el valor de la expresión y el puntero aumenta su valor en:

1. Se repite este proceso con cada expresión.

DEFW expresión, expresión: coloca el valor «expresión» (de 2 bytes) en la dirección del puntero y avanza el puntero en **2**. Se coloca primero el byte menos significativo seguido del más significativo. El proceso se repite en cada expresión.

DEFS expresión

Aumenta el valor del puntero en el número de bytes que «**expresión**» indique. Esto equivale a reservar un bloque de memoria de longitud igual al valor numérico de la expresión. A todas las posiciones de memoria de este bloque se les dará el valor cero.

DEFM «S»

Coloca el valor ASCII de la cadena «**S**» en n bytes, donde n es la longitud de dicha cadena y puede encontrarse, en teoría, en el intervalo entre 1 y 255 inclusive, aunque en la práctica, lo que limita la longitud de la cadena es la longitud de línea que el editor te permitirá introducir. El primer carácter de la zona de operandos (en nuestro caso las comillas) se toma como el delimitador de la cadena y ésta se define como los caracteres existentes entre dos delimitadores; el final de línea también actúa como delimitador.

ENT expresión

El valor de la expresión corresponde a la dirección desde donde el código objeto será ejecutado cuando utilicemos el comando «**R**».

No puede faltar esta dirección de ejecución en el programa.

Pseudonemónicos condicionales

Este tipo de pseudonemónicos permiten al programador quitar y poner determinadas zonas de código fuente para ser o no ensambladas.

IF expresión

Primero se evalúa la expresión; si ésta es cero, no se ensamblarán las líneas siguientes hasta que aparezca «ELSE» o «END» en el programa. Si la expresión no vale cero, el ensamblado continúa normalmente.

ELSE

Cuando el ensamblador encuentra un ELSE, hace lo contrario de lo que venía haciendo. Si no estaba ensamblando, comenzará a hacerlo y viceversa.

END

Fin del ensamblado.

Formato de las líneas

Cada línea de texto que GENA3 **compila** a código máquina debe respetar una sintaxis muy sencilla, al objeto de que el programa sepa a qué atenerse. Las líneas pueden tener el siguiente formato:

ETIQUETA: PRINCI:	NEMONICO LD
OPERANDOS HL, etique	COMENTARIOS ; carga hl con etique

los espacios y tabulaciones (insertadas por el editor) generalmente se ignoran. La línea se procesa de la siguiente forma: Se comprueba si el primer carácter de la línea es: ":", ";", "/*", "*/" o [ENTER].

Se es éste "/*", toda la línea se trata como un comentario, y por tanto ignorada.

Si es éste "/*", el ordenador sabe que el/los siguiente/s carácter/es es un comando del ensamblador. Trata todos los caracteres después del comando como comentario.

Si éste es un [ENTER] simplemente ignora la línea.

Si el primer carácter de la línea se trata de cualquier otro aparte de los listados, anteriormente, entonces el ordenador busca un espacio, una tabulación o dos puntos ":", ";". En este último caso, todos los caracteres precedentes, formarán parte de la etiqueta, de cualquier otro modo, el primer carácter que no sea un espacio o un tabulador se tomará como principio del mnemónico.

Observa que si un mnemónico está a continuación de una etiqueta, sólo los 6 primeros caracteres de la etiqueta aparecerán en la etiqueta. Sin embargo, si una etiqueta está sola en una línea, en el

listado aparecerán todos sus caracteres aunque sólo los 6 primeros entrarán en la tabla de símbolos.

Después de procesar una etiqueta válida, o si no la hay, el ensamblador busca el siguiente carácter que no sea espacio o tabulador, y ve si éste se trata o bien el final de la línea o el comienzo de un mnemónico de Z80 (ver apéndice 2) de hasta 4 caracteres de longitud y terminado por un espacio/tabulador o [ENTER]. Si el mnemónico es válido y requiere uno o más operandos, el ordenador se salta los espacios/tabs y procesa dichos operandos.

Comando de GENA3

De lo que acabamos de ver es importante matizar en la existencia de comandos propios del ensamblador que van precedidos por el símbolo "/*", y que alteran el output de GENA 3 o incluso el propio código de máquina generado. A continuación aparecen listados, indicando su propósito y la función que cumplen:

Comandos del ensamblador

Estos comandos, al igual que las órdenes al ensamblador, no afectan al procesador al ejecutar el programa pues no se convierten en código objeto. Sin embargo, aunque no producen ningún cambio en el código objeto generado por el ensamblador, modifican el formato del listado.

Un comando es una línea de texto fuente que comienza con su asterisco "/*". La letra que sigue determina el tipo de comando debe ser una letra mayúscula. El resto de la línea puede ser cualquier cosa, excepto en los comandos, "L", "T" y "D" que deben ir seguidos de un signo "+" o un signo "-".

Comandos: *E imprime tres líneas en blanco en la pantalla. Es útil para separar partes. En caso de utilizar la impresora, ésta iniciará una nueva página al encontrar este comando.

*Hs imprime la cadena s como encabezamiento después de cada *E. *Hs incluye *E, así que este último comando no será necesario en caso de incluir *Hs.

*S detiene el listado del programa. Este continuará pulsando cualquier tecla. Es útil para leer direcciones de memoria en medio del listado.

*L— a partir de esta línea no se lista el programa.

*L+ a partir de esta línea, el listado aparece en pantalla.

*D+ el valor del puntero se encontrará en base 10 al principio de cada línea, en lugar de su formato normal o hexadecimal.

*D— efecto contrario al comando anterior. El puntero vuelve a aparecer en hexadecimal.

*F nombre del archivo

Este poderoso comando permite ensamblar texto del cassette o del disco. Dicho texto se lee del cassette y se introduce en un buffer, un bloque cada vez y después se ensambla desde el buffer. Esto permite crear grandes cantidades de código objeto puesto que éste no ocupa memoria real. El nombre del archivo (*hasta 8 caracteres*) de texto que quieres incluir en ese momento del ensamblado debe ser especificado después de "+F" y debe ir precedido por un espacio. Si no dices el nombre del archivo, se ensamblará el primero que encuentre el ordenador. Cualquier archivo que quieras incluir de esta manera deberá ser grabado en un cassette usando el comando "P" del editor. Cuando el ensamblador detecte un comando "F" se imprimirá el mensaje "Press PLAY then any key" antes del primero y segundo paso pues el archivo deberá ser revisado en cada piso. El cassette buscará el archivo con el nombre especificado, o en su defecto, el primero de todos. Cuando encuentre uno que no coincida con el que queremos, imprimirá "Found nombre" y si se trata del que queremos incluir: "Loading nombre".

*T+ nombre del fichero

Este comando vuelca código objeto al cassette o al disco (con el nombre dado) durante el ensamblado. Este volcado puede ser interrumpido por el comando *T—, por ORG o por el final del ensamblado. El código así grabado puede ser recargado desde Basic o con MONA3. *T— cierra el archivo de cassette de código objeto. Si el ensamblado ha sido interrumpido por un pseudonmónico condicional, este comando deja sin efecto cualquier otro.

Una de las posibilidades más poderosas de GENA3 es la forma en la que se ensambla el código fuente (*texto*) para producir código objeto (*binario*), es decir, el comando «A». GENA3 posee una serie de opciones de ensamblado, y cada una afecta de manera distinta al código máquina que aparece como resultado de ellas.

Hasta aquí esto no es nada del otro mundo, pero lo interesante es que estas opciones de ensamblaje pueden combinarse por el simple método de sumarlas, esto es, si cuando el programa

pregunta «Options?» yo le digo 5, GENA3 ejecutará las opciones 4 y 1 simultáneamente ($4+1=5$). Las opciones y lo que hacen pueden verse en la tabla número 1.

El editor de GENA3

El editor de pantalla es la herramienta auxiliar del programa ensamblador, y nos permite crear un programa en código fuente y manipular el texto que lo integra de formas muy diversas y potentes.

En mi opinión, el editor está magníficamente pensado (la idea de orientarlo a números de línea, como Basic), y a veces da un poco de pena que el propio editor del Amstrad no incorpore las mismas características, sobre todo en lo que se refiere a búsqueda y sustitución de determinada cadena de caracteres por otra. Yo diría que sólo tiene un fallo, pero grave: no existe la posibilidad de manipular y cambiar de sitio el texto por bloques. Una lástima.

No obstante, una descripción detallada de los comandos del Editor también vale más que 1.000 palabras. Por tanto, ahí va. Comandos del Editor de pantalla de GENA3:

El editor de GENA3 es un sencillo instrumento pero eficiente a la hora de editar programas rápidamente. Para reducir el tamaño del archivo de texto, el editor comprime algunos espacios de la siguiente manera. Cuando tú introduces una línea, ésta se introduce carácter a carácter en un buffer interno. Cuando pulsas [ENTER] se transfiere de buffer al archivo y es durante esta transferencia cuando se comprimen los espacios.

Se busca el primer carácter de la línea; si es un espacio, se introduce un tabulador en el archivo de texto y los demás espacios se saltan. Si no es un espacio, todos los caracteres hasta el primer espacio se introducen en el archivo, y después el primer espacio ocurre lo mismo que si el siguiente carácter fuese el primero de la línea. Esto se repite otra vez, introduciendo tabuladores al principio de la línea, entre

etiqueta y mnemónico y entre mnemónico y operandos.

Si aparece [ENTER], la transferencia finaliza y el control vuelve al editor. Los espacios no se comprimen en los comentarios, ni deberían formar parte de una etiqueta, mnemónico u operando.

El editor genera: el símbolo "<"

En respuesta a este símbolo (que indica que el ordenador se encuentra preparado para recibir órdenes), puedes introducir una línea de comandos con el siguiente formato:

C N1, N2, S1, S2 [ENTER]

C es el comando a ejecutar

N1 número entre 1 y 32767 ambos inclusive

N2 número entre 1 y 32767 ambos inclusive

S1 cadena de caracteres de hasta 20 caracteres de longitud

S2 cadena de caracteres de hasta 20 caracteres de longitud

Las comas se usan para separar los diversos argumentos (aunque esto puede cambiarse —ver comando "S") y los espacios son ignorados, excepto en las cadenas. Ninguno de los anteriores argumentos son obligatorios, aunque, por ejemplo, el comando "D" no funcionará si no se especifican N1 y N2. El editor recuerda los últimos números y cadenas que introdujiste y los utiliza si no especificas alguno de ellos. El valor inicial de N1 y N2 es 10, y de S1 y S2 la cadena vacía. Si introduces una línea no válida como: F-1, 100, HOLA, la línea será ignorada y se imprimirá el mensaje "Pardon?" y deberás introducirla correctamente: F2, 100, HOLA. Este mensaje se imprimirá también si la longitud de S2 es mayor que 20. Si S1 tiene más de 20 caracteres, el sobrante será ignorado. Los comandos pueden introducirse en mayúsculas o minúsculas.

[CTRL]X borra hasta el principio de la línea.

[TAB] desplaza el cursor hasta la posición de la siguiente tabulación las siguientes sub-secciones detallan los comandos existentes en el editor. Nótese que cuando un argumento está encerrado entre los símbolos "< +", es necesario para que el comando tenga efecto.

Comandos del editor

El texto puede introducirse, o bien escribiendo el número de línea, un espacio y después el texto o usando el comando "I". Si introduces un número de línea seguido de [ENTER], ésta será borrada

del programa si existía previamente.

[CTRL]C vuelve al editor y ya conocemos las funciones de [CTRL]X y [TAB]. La tecla [DEL] borra un carácter (no más allá del principio de la línea). Si el buffer en el que se introduce la línea se llena, el ordenador te impedirá seguir escribiendo, debiendo utilizar [DEL] o [CTRL]X para hacer hueco. Si, durante la sección de texto, aparece el mensaje "Bad Memory!", esto querrá decir que nos estamos acercando al final de la RAM, no se puede introducir más texto y el programa en cuestión (o al menos parte) deberá ser grabado para posterior recuperación.

Comando: I n, m

Modo de inserción automático. Los números de línea aparecen automáticamente empezando por el número n y con incrementos de m.

Para salir de este modo pulsa [CTRL]C.

Se introduces una línea con un número que existía previamente, la anterior línea será remunerada una unidad más y la nueva línea insertada antes, después de pulsar [ENTER]. Si el incremento automático de n.º de línea produce una línea con un número mayor de 32767, el ordenador saldrá de este modo automáticamente.

Listado de textos

Comando: L, n, m

Lista el programa desde la línea n hasta la m. Si no se especifican, n tomará el valor de 1, y m el valor 32767, con lo cual se listará todo el programa. La tabulación es automática y el número de líneas que caben en la pantalla es 24. Cuando haya listado 24, el editor hará una pausa. Pulsa [CTRL]C para volver al editor o cualquier otra tecla para continuar.

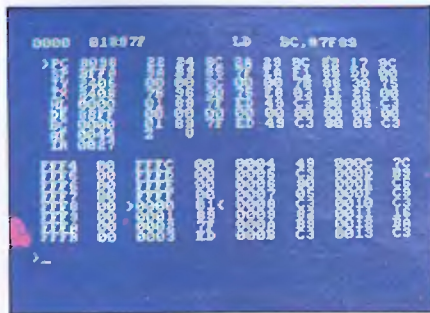
Edición de textos

Comando: N <n,m +

Renumerar el programa, dando el número n a la primera línea n+m a la segunda. n+m+m a la tercera, y así sucesivamente. Si al renumerar, sobrepasamos con algún número de línea el valor 32767, se conservará la numeración antigua.

Comando: F n, m, f, s

busca la cadena f en todas las líneas desde n hasta m. Si f aparece en alguna de ellas, ésta será impresa y el ordenador se introduce en el modo Editor automáticamente (ver siguiente comando). Puedes, entonces, usar los comandos del Editor para buscar otra vez la cadena f en el intervalo de instrucciones dado,



sustituirla por la cadena S. (n, m, f, s pueden haber sido dadas anteriormente, y por tanto no ser necesario escribirlas de nuevo para hacer la misma búsqueda posteriormente. Bastará para ello simplemente con introducir "F".

Comando **E n**

Edita la línea de número n. Si no existe, no ocurrirá nada, pero si existe, la línea será adoptada en un buffer e impresa en pantalla.

El número de línea se imprimirá inmediatamente debajo. Todos los cambios a partir de aquí tendrán lugar en el buffer, no en el programa mismo, siendo posible recuperar la línea original en cualquier momento. En este modo, un puntero se coloca al principio en el primer carácter de la línea y varios subcomandos permiten arreglar los errores. Estos son:

"+" (espacio) incrementa el puntero en uno. No seguirá retrocediendo más allá del principio de la línea.

[DEL] coloca el puntero en el carácter anterior. No seguirá retrocediendo más allá del principio de la línea.

[ENTER] finaliza la edición de la línea guardando los cambios hechos.

Q - sale del modo de edición sin tener en cuenta los cambios hechos.

R - ignora los cambios hechos y recarga el buffer con la línea como estaba originalmente.

L - lista el resto de la línea desde la posición del puntero. Permaneces en el modo de edición con el puntero al principio de la línea.

K - borra el carácter en el que se encuentra el puntero.

Z - borra todos los caracteres desde la posición del puntero hasta el final de la línea.

F - explicado anteriormente.

S - sustituye la cadena f encontrada por "F" por la cadena s y continúa la búsqueda de f más adelante.

I - inserta caracteres en la posición del cursor. Continuarás en este submodo hasta que pulses **[ENTER]**, tras lo cual volverás al modo editor con el puntero colocado en el último carácter que has insertado. **[DEL]** borra el carácter a la izquierda del puntero mientras estemos en el submodo I (en este submodo el cursor será un asterisco "***").

X - avanza el puntero hasta el final de la línea e introduce el modo I.

C - cambia el carácter en el que se encuentre el puntero por el que tu escribas e incrementa el puntero en uno.

Permanecerás en este modo hasta pulsar **[ENTER]**, tras lo cual volverás al editor colocado en el último carácter que cambiaste. **[DEL]** hace retroceder el puntero una posición y **[TAB]** no tiene efecto. El cursor en este submodo es el signo "+".

Comandos del cassette o disco

Los programas pueden ser grabados y cargados usando los comandos "P", "V" y "G" mientras que el código objeto será grabado mediante el comando "O".

Comando: **Pn, m, s**

El intervalo de líneas entre n y m se graba bajo el nombre s. Recuerda que estos argumentos pueden tener valores asignados previamente. Asegúrate de que el cassette está en modo RECORD antes de introducir este comando.

Comando: **Q,,s**

Es igual que el comando P pero el texto se graba en ASCII sin números de línea y con **CRLF** (retorno de carro/fin de línea) al final de cada línea.

Comando: **G,,s**

El cassette busca un archivo con el nombre s, si lo encuentra, será cargado detrás del programa que esté en memoria. Si s es la cadena vacía, se cargará el primer programa que aparezca.

Después de introducir el comando "G", pulsa PLAY en el cassette. Si encuentra el programa deseado el mensaje "Loading nombre" aparecerá, si no, el mensaje será "found nombre" y la búsqueda continúa. Si hay algún otro programa en memoria, el programa cargado con "G", se añadirá al final del anterior y ambos serán renumerados desde la línea 1, con intervalo de 1 entre líneas y línea.

Comando: **V,,s**

verifica el programa s del cassette con el existente en la memoria. Si la verificación es correcta, el mensaje que aparece es "Verified" si no obtendremos el mensaje "Failed!"

Comando: **O,,s**

graba el código objeto generado en el último ensamblado bajo el nombre s. Para cargar este código objeto utiliza MONA3 o BASIC.

Comando: **Tn**

Cambia la velocidad de grabación en cassette. "T" seguido de **[ENTER]** selecciona la velocidad lenta, mientras que "T" seguido de un número mayor que cero y **[ENTER]** selecciona el modo rápido.

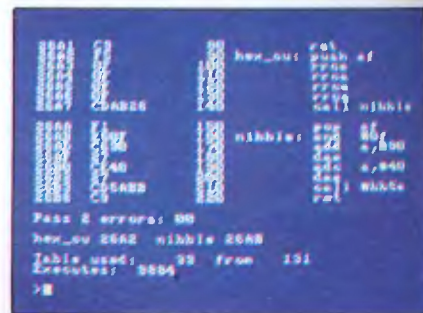
Ensamblado y ejecución desde el editor

Comando: **A**

Ensambla el programa fuente desde la primera línea.

Comando: **R**

Si el código fuente ha sido ensamblado sin



errores, y has dado la dirección de ejecución con ENT, el comando "R" ejecutará el código objeto. Este programa puede incluir una instrucción RET (C9) para volver al editor puesto que el stack se encuentra en la misma situación al finalizar la ejecución que al principio. ENT no tendrá ningún efecto si has escogido la opción 16 al ensamblar.

Otros comandos

Comando: **H**

Imprime una pantalla de ayuda donde aparecen los comandos de GENA3 en mayúsculas.

Comando: **B**

Devuelve el control al sistema operativo. Para reentrar en el ensamblador, puede hacer una llamada a la dirección original de carga +2, en cuyo caso borrarás el programa fuente de la memoria o bien llamar a la dirección original +4, conservándolos.

Comandos: **S,,d**

Te permite cambiar el delimitador que se toma para separar argumentos en una línea. Al entrar en el editor, se toma la coma "," como delimitador, pero tas utilizar este comando, el papel de delimitador pasará al primer carácter de la cadena d, y hasta que no se defina otra vez, dicho carácter será el delimitador. No se puede utilizar el espacio para este fin.

Comando: **C**

Imprime el valor del delimitador, N1, N2, S1 y S2. Es útil para comprobarlos en caso de que pienses omitir alguno de ellos en algún comando.

Comando: **Zn, m**

Lista en impresora el trozo de programa entre las líneas n y m. Si ambos valores se omiten, todo el programa fuente será listado. En caso de que la impresora no esté conectada, se imprimirá el mensaje "No Printer!" y no ocurrirá nada.

Puede parar el listado pulsando tecla. Si pulsas **[CTRL]C**. Volverás al editor, mientras que pulsando cualquier otra cosa, el listado continuará.

Comando: **Yn**

Pone el número de líneas por página al valor n. Esto te permite ajustar la salida



por impresora a las diferentes longitudes de papel de impresora.

Comando: **X**

Imprime la dirección de comienzo y la final del código fuente en decimal.

Es útil en caso de que quieras grabar el texto desde Basic o para ver cuánta memoria tienes libre tras dicho programa "**X**" se usa también en conjunción con el desensamblador que MONA3 para desensamblar un programa para uso de GENA3. Para unir un texto generado por MONA3 al ensamblador, debemos mover primero dicho texto (o programa en código fuente) al TEXTSTART de GENA3 o generar el texto directamente en esa dirección.

Ahora, la primera dirección impresa con el comando "**X**" corresponde al principio del texto de GENA3. Esta es la dirección en la que el código fuente desensamblado por MONA3 debe empezar, así que habrás de ingeniártelas para desensamblar el código en esa posición y anotar la dirección del final del texto que te da el desensamblador, valor que deberás dar al TEXTEND de GENA3. Este TEXTEND está almacenado en la dirección "**Principio de GENA3**" + 7 así que si cargamos GENA3 en 1000, TEXTEND estaría localizado en 1007. Toma el valor del final del texto dado en hexadecimal por MONA3 y "**poke**" el byte menos significativo en 1007 y el más significativo

en 1008 y llama a GENA3 vía "**dirección original**" + 4. El programa desensamblado puede ser editado y ensamblado por GENA3 ahora.

Comando: **U**

Muestra el número de línea de la última línea del fichero de texto, es útil para encontrar rápidamente dónde termina el fichero de texto y añadir más texto.

Comando: **W**

Cambia el display de 40 a 80 caracteres o al revés.

Comando: **[CTRL/J]**

Te introduce en el programa para descubrir errores MONA3 si éste ha sido utilizado previamente, si no, no tiene ningún efecto.

Respecto al GENA3, esto es todo. Pero el ensamblador no estaría completo si no viniera acompañado de otro paquete-herramienta que cumple la misión contraria: un desensamblador-monitor también muy potente que nos permite depurar un programa lenguaje máquina hasta límites verdaderamente sorprendentes. Lo analizaremos la próxima semana.

Cuando utilices el comando "**A**" para ensamblar se te pedirá el tamaño de la tabla (**table size?**) en decimal. Esto representa el espacio que se le da a la tabla de símbolos durante el ensamblado. Si no pones nada (si pulsas **[ENTER]**), entonces GENA3 escogerá una longitud para la tabla que considere suficiente. Normalmente, este espacio será bastante. Cuando escojas la opción de incluir archivos, puede ser necesario que especifiques el tamaño de la tabla, pues el ensamblador no puede predecir la longitud del fichero que vas a añadir.

Después de "**Table size?**" se te pedirá la opción que deseas utilizar cuando aparezca el mensaje "**Options:**" Veamos cada una de ellas:

OPCIONES DE GENA3 PARA ENSAMBLAR

OPCION 1	Produce un listado de la tabla de símbolos después del segundo paso del ensamblado.
OPCION 2	No produce código objeto. Es decir, no genera Código Máquina (<i>sirve para descubrir errores en el programa sin ensamblar realmente cuando no tenemos espacio en memoria para el programa en Código Máquina</i>).
OPCION 4	No lista el programa ensamblado.
OPCION 8	Lista el programa ensamblado en impresora.
OPCION 16	Coloca el código objeto después de la tabla. El puntero se coloca en la dirección señalada por ORG. de manera que el Código Máquina puede ser puesto en una zona de memoria pero ejecutado en cualquier otro sitio.
OPCION 32	No revisa la zona de memoria en la que el código objeto va a ser colocado. Es útil para dar mayor rapidez al ensamblado.

Tabla 1.

GANAR
100.000 PESETAS
CON MICROHOBBY

AMSTRAD

SEMANAL

Porque pretendemos que **AMSTRAD SEMANAL** sea también vuestra revista, hemos abierto una sección en la que se publicarán los mejores programas originales recibidos en nuestra redacción. Vosotros seréis los encargados de realizar estos páginos, en las que podréis aportar ideas y programas interesantes para otros lectores.

Las condiciones son sencillas:

- Los programas se enviarán a **AMSTRAD SEMANAL** en una cinta de cassette, sin protección en el software, de forma que sea posible obtener un listado de los mismos.
- Cada programa debe ir acompañado de un texto explicativo en el cual se incluyan:
 - Descripción general del programa.
 - Tabla de subrutinas y variables utilizadas, explicando claramente la función de cada una de ellas.
 - Instrucciones de manejo.
- Todos estos datos deberán ir escritos a máquina o con letra clara para mayor comprensión del programa.
- En una sola cinta puede introducirse más de un programa.
- Una vez publicado, **AMSTRAD SEMANAL** abonará al autor del programa de **15.000 a 100.000** pesetas, en concepto de derechos de autor.

— Los autores de los programas seleccionados para su publicación, recibirán una comunicación escrita de ello en un plazo no superior a dos meses a partir de la fecha en que su programa llegue a nuestra redacción.

— **AMSTRAD SEMANAL** se reserva el derecho de publicación o no del programa.

— Todos los programas recibidos quedarán en poder de **AMSTRAD SEMANAL**.

— Los programas sospechosos de plagio serán eliminados inmediatamente.

¡ENVÍANOS TU PROGRAMA!

a **HOBBY PRESS, S. A. La Granja, n.º 8.**
Pol. Ind. Alcobendas (Madrid)

Sin duda alguna

A través de esta sección se pretende resolver, en la medida de lo posible, todas las posibles dudas que «**atormenten**» a todas las personas interesadas en el mundo del AMSTRAD, sean o no poseedores de uno y, si lo son, se encuentren en cualquier nivel de destreza en su manejo.

Semanalmente, aparecen en estas páginas las consultas de la mayor cantidad de usuarios posible; ello redundará en un mejor servicio y en un contacto más estrecho entre todos nosotros a través de la revista.

SIN DUDA ALGUNA está abierta a todos.

Ante todo he de felicitaros por la acertada decisión de crear esta vuestra/nuestra revista MICROHOBBY **AMSTRAD**. Es una publicación excelente y muy dinámica. Continúa así, por favor.

Tengo un serio problema. Necesito disponer de la letra ñ, en minúscula y mayúscula, para mi CPC-664. Al comprar el ordenador me informaron que la tecla marcada con el símbolo «\», situada justo encima de la de control, es la que está destinada para crear dicha letra. Esto es lo que sucede cuando se utiliza el programa «Animal, Vegetal y Mineral» que regalan al comprar el ordenador. ¿Cómo podría yo programar dicha tecla para que al pulsarla obtenga en pantalla la ñ?

Mil gracias por anticipado.

Manuel Delgado (Vitoria)

En el número 7 de **AMSTRAD SEMANAL** publicábamos una simpática carta de uno de nuestros lectores en la cual resolvía tu problema, dando un corto programa Basic que convierte una de las teclas del ordenador en la letra que necesitas, además de explicar las razones por las que lo hacía.

Soy el orgulloso propietario de un **Amstrad CPC464** desde hace un par de semanas y aprovechando la agradable sorpresa de la aparición de su revista quiero hacerles unas preguntas:

1. En la guía del usuario que acompaña al ordenador se puede leer que no se debe experimentar con el POKE. ¿Cómo podría un inexperto como yo saber las diferentes funciones de las distintas posiciones de memoria? Me pregunto lo mismo sobre CALL.

2. ¿Cuántos gráficos SPRITES pueden usarse en el 464 y cómo se realizan?

3. En el manual dice que existen 32 teclas programables comprendidas entre los caracteres ASCII 128 y 159, pero yo sólo he logrado programar del 128 al 139, es decir, el teclado numérico reparado. ¿Cuáles son los otros?

Vicente Matellán / León.

1. La razón de que el manual advierta contra el uso del comando POKE se debe a que el **Amstrad** mantiene partes críticas de su sistema operativo en memoria de lectura —escritura (RAM), en unas zonas a las que tú puedes acceder. Si cambias inadvertidamente alguno de estos números con POKE, el ordenador no funcionará correctamente y se bloqueará.

Para saber exactamente qué zonas de memoria no debes manipular, o al menos qué hay allí, debes leer con atención el libro del Firmware de **Amstrad**. El mapa de memoria que viene en el manual también te ayudará algo al principio. No hay otra manera.

2. El **Amstrad CPC 464** no posee gráficos Sprites. Lo más que puedes hacer es simular el efecto Sprite definiendo tu propio juego de caracteres y usando las sentencias que controlan las interrupciones, AFTER y EVERY.

3. Las otras teclas a las que aludes pueden programarse mediante la sentencia KEY DEF, seguida de una serie de parámetros. Por razones de espacio nos vemos obligados a omitir una descripción completa.

AVISO A NUESTROS LECTORES

Comunicamos a todos nuestros lectores que a partir del día 1 de enero de 1986 nuestro Departamento de Clientes será trasladado al Polígono Industrial de Alcobendas, C/ La Granja, 39. Alcobendas (Madrid). Para cualquier consulta que deseen realizar pueden llamar a los teléfonos **654 28 98** y **654 28 67** de 9.00 a 14.30 h. y de 16.00 a 18.30 h.



¡Operación cambio!

Valoramos:

Tu AMSTRAD 464 en 50.000 ptas.

Un Spectrum+ en 30.000 ptas.

Amstrad CPC en 70.000 ptas.

En la compra del AMSTRAD CPC 6128 o PCW 8256.

Consulte para monitor color.

Precios especiales en impresoras y accesorios.



270 34 97.



New Line

GABINETE DE INFORMÁTICA

- **Clases de Informática sobre AMSTRAD**
EN GRUPOS O INDIVIDUALES
- **Ordenadores AMSTRAD y periféricos**
Los mejores precios
- **Software a la medida**

ZURBANO, 4. ☎: 410 47 63
28010 MADRID

Amstrad Ideas

AMSTRAD Semanal comunica a todos sus lectores la apertura de una nueva sección dedicada a recoger las mejores ideas que exploten al máximo las posibilidades del ordenador, materializadas en programas claros y cortos (máximo 25 líneas). Los mejores de entre todos ellos serán publicados con el nombre de su autor en la revista, recibiendo como premio, gratuitamente en su domicilio los cuatro primeros números de nuestra cinta mensual. Los programas enviados deberán incluir:

- **Cinta de cassette con el programa o programas grabados.**
- **Explicación detallada del funcionamiento y propósito del programa, mecanografiado a 2 espacios o con letra clara.**

Es imprescindible indicar en el sobre claramente: **AMSTRAD IDEAS**.

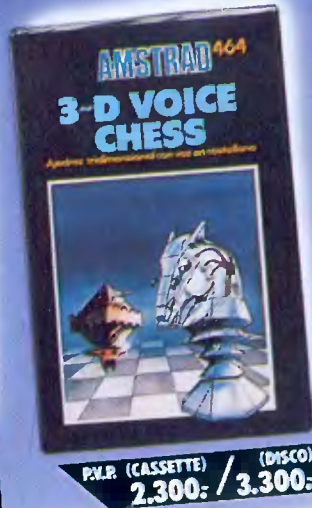
La dirección es:

Hobby Press, S. A.

La Granja, s/n.

Polígono Industrial de Alcobendas.
Madrid

AMSTRAD SOFT



ACE

DISTRIBUCION

Actividades Comerciales Electrónicas, S.A.
Tarragona, 112 Tel. 325 15 12* Telex 93133 ACEE E 08015 Barcelona

YA DISPONIBLE EN

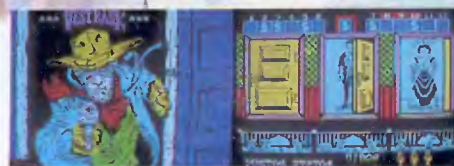


... Y EN TODAS LAS
TIENDAS ESPECIALIZADAS

WEST BANK



Defiende el
banco de Soft City
del ataque de los forajidos
y consigue escribir tu nombre
en la leyenda del «FAR WEST».



Fotos tomadas de un Spectrum 48 K

SPECTRUM · 1950 · AMSTRAD

(DISK · 3050)

**PEDIDOS CONTRA
REEMBOLSO:
TEL.: (91) 715 00 67
TIENDAS Y
DISTRIBUIDORES:
TEL.: (91) 447 34 10.**

DINAMIC

**«MANSION DINAMIC»
C/ TILOS, 2, 21,
MONTEPRINCIPE
BOADILLA DEL MONTE
MADRID**